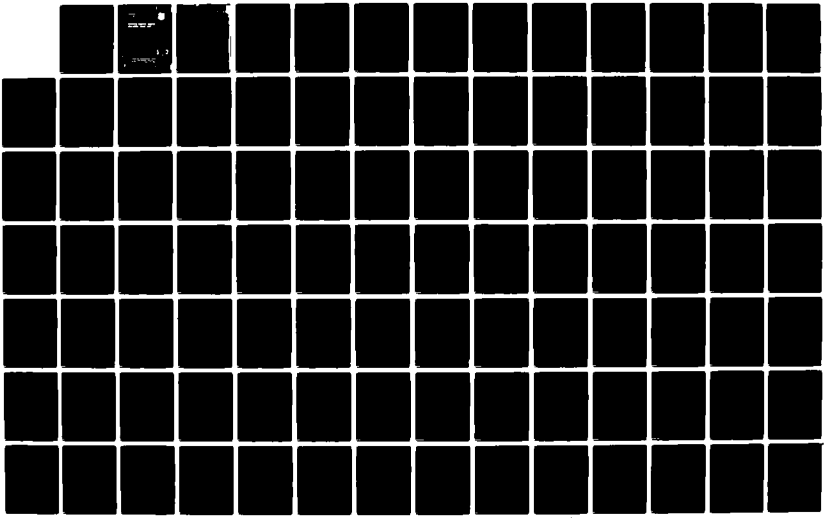
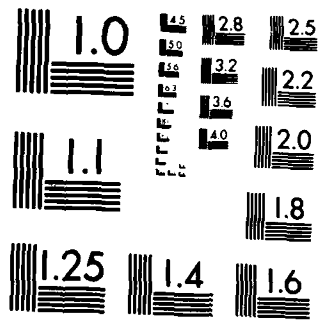
AD-A125 626 INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY 1/5
VOLUME 1(U) GENERAL DYNAMICS FORT WORTH TX FORT WORTH
DIV H C CONN ET AL. JAN 83 DMA-2-014-VOL-1

UNCLASSIFIED RADC-TR-83-3-VOL-1 F30602-81-C-0039 F/G 9/2 NL



M-2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD A125626

INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY

General Dynamics Corporation

R. C. Camp, Jr, D. J. Rodjak, M. A. Goode, R. M. Bond,
C. O. Anderson, R. C. Robertson

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

83 03 15 004

APPROVED: *Roger B. Panara*

ROGER B. PANARA
Project Engineer

APPROVED: *John A. Beyerle*

JOHN A. BEYERLE, Lt Colonel, USAF
Acting Chief, Command and Control Division

FOR THE COMMANDER: *John P. Huss*

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (COEE) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-83-3, Vol I (of three)

2. GOVT ACCESSION NO.:

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY

5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, 6 Jan 81 - 30 Sep 82

6. PERFORMING ORG. REPORT NUMBER: DMA-2-014

7. AUTHOR(s):

H.C. Conn, Jr.
D.J. Rodjak
M.A. Goode
R.M. Bond
C.G. Anderson
R.C. Robertson

8. CONTRACT OR GRANT NUMBER(s): F30602-81-C-0039

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
General Dynamics/DSD/Central Center
North Grant Lane
Ft Worth TX 76108

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63701B, 32050326

11. CONTROLLING OFFICE NAME AND ADDRESS:
Rome Air Development Center (COEE)
Griffiss AFB NY 13441

12. REPORT DATE: January 1983

13. NUMBER OF PAGES:

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report):
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
Same

18. SUPPLEMENTARY NOTES:
RADC Project Engineer: Roger Panara (COEE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
Software Engineering
programming environment
software tools

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

Volume I (of three) describes the development of the design and supporting documentation for an incremental and evolving integrated modern engineering software production environment for the Defense Mapping Agency.

Volume II is the System/Subsystem Specification.

Volume III is the Functional Description.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FINAL REPORT
PREFACE

This document is the Final Report, CDRL A008, produced as part of the Interactive Computer Program Development System Study for the Defense Mapping Agency. An Executive Summary is provided at the beginning of the report to provide a concise description of the major aspects of the study. The tools and equipment recommended as a result of this study are the ones which best satisfied the requirements and constraints of the Defense Mapping Agency (DMA) environment at the time this document was produced.

PRECEDING PAGE BLANK-NOT FILMED

FINAL REPORT
TABLE OF CONTENTS

FINAL REPORT
TABLE OF CONTENTS (CONT.)

FINAL REPORT

TABLE OF CONTENTS (CONT.)

FINAL REPORT
FIGURES

FINAL REPORT
FIGURES (cont.)

EXECUTIVE SUMMARY
for the
DEFENSE MAPPING AGENCY
MODERN PROGRAMMING ENVIRONMENT
SPECIFICATION

I. Abstract

This summary provides a synopsis of the technical requirements specification, a cost estimate, and an implementation schedule plan for a Modern Programming Environment (MPE) for the Defense Mapping Agency (DMA). The conclusions and recommendations stated in this summary are the results of the Interactive Computer Program Development System Study performed by General Dynamics Data Systems Division (GD/DSD) under contract F30602-81-C-0039 to Rome Air Development Center (RADC). The objectives of this study were:

1. To identify DMA needs for a Modern Programming Environment.

2. To formulate a total systems concept to satisfy the identified needs.

3. To survey and evaluate software tool candidates for the Modern Programming Environment.

4. To specify a Modern Programming Environment and an implementation plan that satisfies DMA needs.

The study was conducted with full cognizance of both recent in-house DMA activities such as the Software Improvement Program (SIP), and currently contracted system development efforts; for example, the Digital Stereo Comparator Compiler, TES/EMPS, Universal Rectifier, and the Clustered Carto System. The study conclusions and recommendations are compatible with these in-house and contracted efforts.

The primary contract deliverables are three reports: (1) a Functional Description of the MPE, (2) a System/Subsystem Specification that details the MPE configuration and identifies particular software tools, and (3) a Final Report that summarizes all stages of the study and provides cost and schedule estimates.

The recommended MPE configuration is a network of VAX-11/780 computers that support ANSI FORTRAN and COBOL software

lifecycle tool environments for software development and maintenance. This VAX network has telecommunication links with production mainframe and individual minicomputer targets. At the time of the preparation of this report, the VAX-11/780 computer is the state-of-the-art technology that best satisfies the requirements of the DMA MPE. Therefore, the VAX-11/780 computer will be referenced as the tool bearing host throughout this report. The MPE configuration can easily support the Ada* language should the Defense Mapping Agency employ Ada in future work. It is recommended that a full MPE configuration be duplicated at both DMAHTC and DMAAC to enhance the utilization of common software between centers.

It is estimated that the total cost (hardware [illegible] software procurements plus contractor development) for im[illegible]mentation of the DMA MPE is $11 million. These funds [illegible] expended during a 54 month implementation period that star[illegible] in fiscal year 1983 and ends in fiscal year 1988. The im[illegible]ntation plan consists of four phases:

1. Phase I - Near-term experimental system
2. Phase IA - Near-term full-scale system
3. Phase II - Far-term experimental system
4. Phase IIA - Far-term full-scale system

This implementation plan includes DMA decision points for continuation of work authorization and assures a working system is available at the end of each phase. The outlook is for experimental near-term capabilities to be available in 1985 and then evolving to full far-term capabilities in 1987.

The benefits of the Modern Programming Environment to the Defense Mapping Agency are twofold--cost and technical capability. First, it is estimated that the entire $11 million implementation cost is recovered within five years (in fiscal year 1988) from the start of implementation and in five more years (in fiscal year 1993) the cumulative net savings of the Modern Programming Environment is $25 million. Secondly, the Modern Programming Environment provides the tools, methodologies, and guidelines to meet the increasing strategic and tactical requirements for the processing of digital data which would be impossible to meet using existing methods.

* Ada is a registered trademark of the U.S. Government (AJPO)

II. Technical Summary of the Modern Programming Environment

There are several components to a modern programming environment. The relationships among these components can be represented by a "layered model" wherein the set of all interior layers supports the next outermost layer. The particular components and layering for the recommended Defense Mapping Agency Modern Programming Environment are shown in Figure I. It is recommended that both DMAHTC and DMAAC have this MPE configuration to facilitate the use of common software.

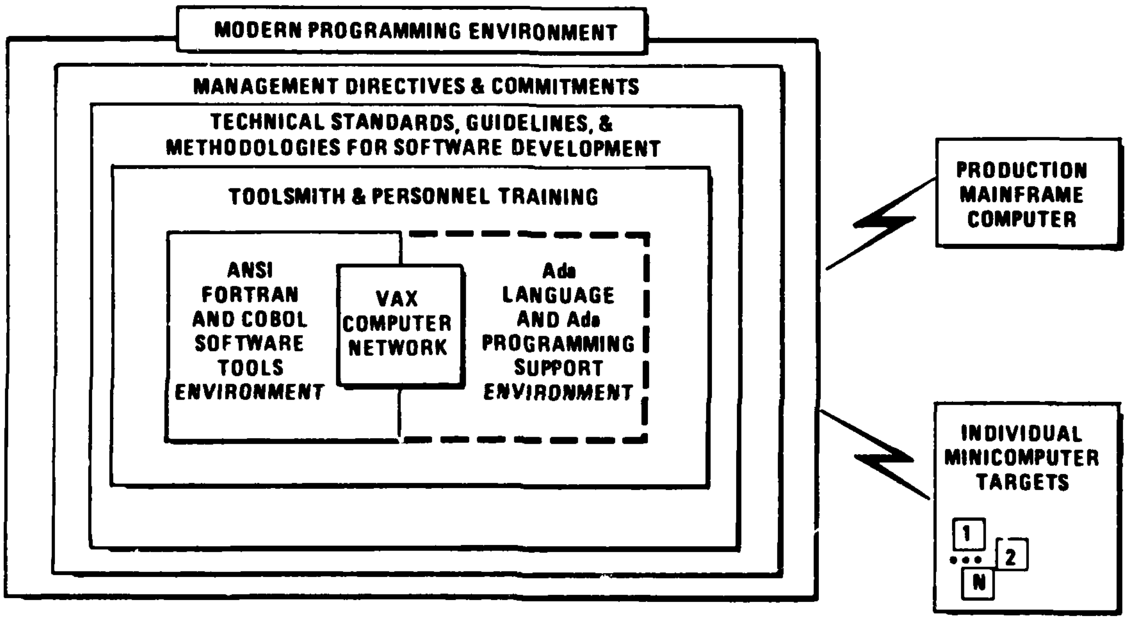


Figure I: DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT CONFIGURATION

The core of the DMA MPE is a network of VAX-11/780 computers that serves as the tool bearing host for the FORTRAN and COBOL software tool environments. The Ada environment is being developed under tri-services sponsorship, and it should be available to the DMA MPE as a government-owned environment if DMA uses Ada in future work. This situation is represented by the dotted line around the Ada portion of Figure I. The principal factors that led to the selection of the VAX-11/780 as the tool bearing host are:

1. A full complement of lifecycle tools that supports DMA's needs for software development and maintenance already exists.

2. Vendor support for the VAX system is excellent. Much government, military, and commercial software R&D efforts are already targeted to the VAX and more such efforts are expected in the future. Examples of organizations that already have VAX based system development environments include: Bell Research Labs, TRW, Air Force Wright Aeronautical Laboratory, General Research Corporation and Boeing. Therefore, DMA can upgrade their MPE tool set in the future at little or no additional cost.

3. The cost and facilities requirements for a VAX system are considerably less than a mainframe computer tool bearing host.

4. The DMA already has several contracted efforts, namely, TES/EMPS, Clustered Carto, and PAMS that are based on a VAX-11/780 system. Hence, the maintenance of these contractor developed systems by DMA using a compatible VAX based MPE will be very cost effective.

Table I shows the recommended set of tools for the DMA MPE that supports the complete software lifecycle; that is, the requirements, design, coding, testing, and maintenance phases as well as the project management and training activities. These tools constitute the second layer in Figure I.

SOFTWARE TOOL		SUPPORTS LIFE CYCLE PHASE FUNCTIONS
•USE.IT	– User System Evaluation and Integration Tool	Requirements, Design, and Automated FORTRAN Coding
•SDDL	– Software Design and Documentation Language	Design
•DMATRAN/IFTRAN	– Defense Mapping Agency Structured Higher Order Language	Coding
•FORTRAN 77	– ANSI Standard Higher Order Language for Scientific Applications	Coding
•COBOL 74	– ANSI Standard Higher Order Language for Business Applications	Coding
•Ada	– Department of Defense Standard Higher Order Language	Coding
•FAVS/RXVP80	– FORTRAN Automated Verification System	Testing
•CAVS	– COBOL Automated Verification System	Testing
•IS/1	– Interactive System/One Programmers' Workbench	Maintenance · Documentation, Text Editing, Configuration Control
•APSE	– Ada Programming Support Environment	Lifecycle Development and Maintenance
•VUE	– Project Management and Critical Path Method Scheduling System	Project Management
•HYPERGRAPHICS	– Microcomputer Based System for Preparation and Presentation of Textual and Graphic Material (rehost to VAX in far-term)	Training

TABLE I: DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT SOFTWARE TOOLS

Each of these tools wc.ld be hosted on the VAX in the far-term; consequently, the DMA MPE has the benefit of being a largely stand-alone software development and maintenance facility.

To describe "how to" effectively use the capabilities of the DMA MPE, the particular sequence and conditions in which the individual software tools would be used has been modeled. This model included five software development and maintenance scenarios:

Scenario 1 - Maintenance of existing software which has not been Software Improvement Program (SIP) upgraded.

Scenario 2 - Maintenance of existing software which has been SIP upgraded.

Scenario 3 - Software presently under development for which standards were not specified.

Scenario 4 - New software to be developed by DMA for which standards will be specified.

Scenario 5 - New software to be developed by contractors for which standards will be specified.

An overview of the descriptions of the software tool use that is common to all of these scenarios is shown in Figure II.

Figure II: OVERVIEW OF SOFTWARE TOOL USE FOR DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT

Two important features of the detailed scenario descriptions are: (1) all software lifecycle phases are supported by automated tools and (2) there are only two tool approaches--a conventional tools approach and an automatic programming approach. These scenarios will form the basis for the toolsmith and personnel training layer of Figure I.

Detailed methodologies, software development standards and guidelines, and training course development will be part of the planned implementation follow-on task and are not contained in this document. However, the draft versions of

the DMA Software Life Cycle Standards prepared under the auspices of the SIP effort are entirely compatible with this MPE specification. It is expected that continued co-ordination between SIP and MPE will result in a set of Software Life Cycle Standards that is supported by the MPE capabilities and conversely.

The principal technical benefits of the recommended DMA MPE are:

1. The stand-alone characteristic of the VAX network plus software tool complement will permit easy training of personnel and high programmer productivity in software development and maintenance efforts.

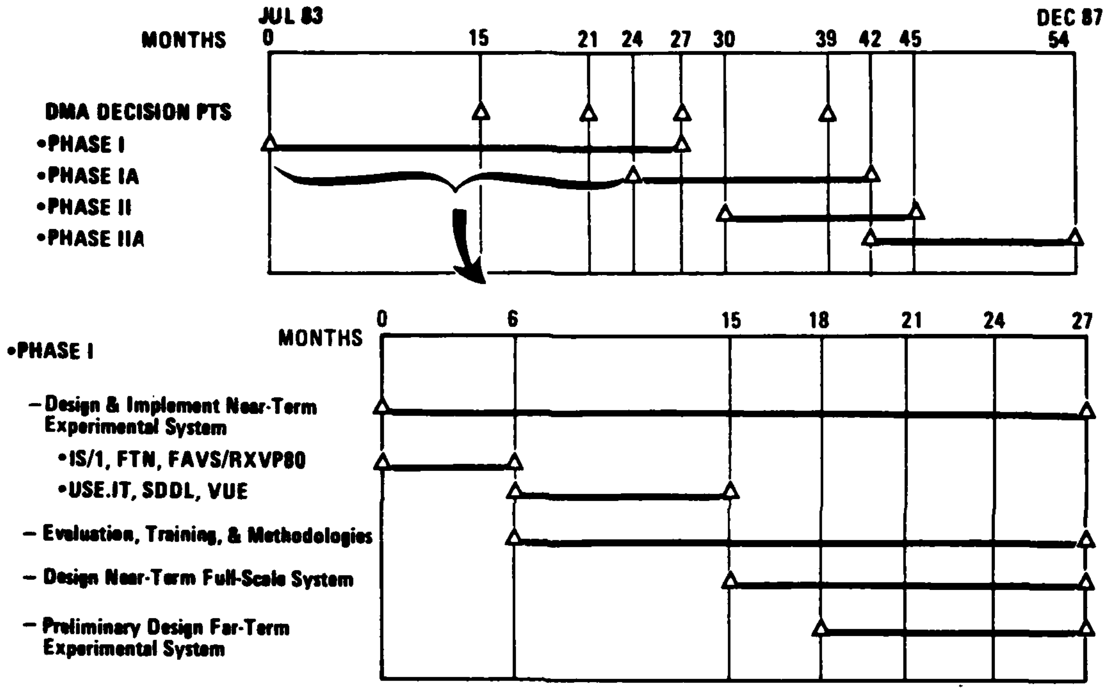
2. State-of-the-art software tools are available now for the VAX and the trend is to continue tool developments for VAX systems.

3. The network capability permits easy growth as DMA processing requirements increase.

4. The VAX based MPE is inherently compatible with several new systems that are now under development. Software maintenance of these systems using the MPE will be facilitated.

III. Schedule and Cost Estimates for the Modern Programming Environment Implementation

The recommendation for the implementation of the DMA Modern Programming Environment as specified in this document is a four-phased program spanning 54 calendar months (July 1983 to December 1987). The schedules and tasks for each phase are shown in Figure III.



PHASE	TASKS
I	•DESIGN, IMPLEMENTATION, & TRAINING FOR NEAR-TERM EXPERIMENTAL SYSTEM •DESIGN NEAR-TERM FULL-SCALE SYSTEM •DEVELOP METHODOLOGIES •PRELIMINARY DESIGN OF FAR-TERM EXPERIMENTAL SYSTEM
IA	•IMPLEMENTATION OF NEAR-TERM FULL-SCALE SYSTEM •NETWORK VAXs AND LINK TO MAINFRAME •TRAINING ON NEAR-TERM FULL-SCALE SYSTEM •UPGRADE METHODOLOGIES
II	•DESIGN AND IMPLEMENT FAR-TERM EXPERIMENTAL SYSTEM •DESIGN FAR-TERM FULL-SCALE SYSTEM •INTEGRATE SOFTWARE TOOLS •UPGRADE METHODOLOGIES •IDENTIFY R&D EFFORTS
IIA	•IMPLEMENT FAR-TERM FULL-SCALE SYSTEM •TRAINING ON FAR-TERM FULL-SCALE SYSTEM •FINALIZE METHODOLOGIES

Figure III: TASKS AND SCHEDULES FOR DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT IMPLEMENTATION

This four-phased schedule plan has several benefits to DMA, in particular:

1. There is low risk because of the careful interleaving of implementation and design tasks within each phase as shown in Table II. An implementation task is always preceded by its corresponding design task.

PHASE	IMPLEMENTATION TASK	DESIGN TASK
I	NEAR-TERM EXPERIMENTAL SYSTEM	NEAR-TERM FULL-SCALE SYSTEM FAR-TERM EXPERIMENTAL SYSTEM
IA	NEAR-TERM FULL-SCALE SYSTEM	
II	FAR-TERM EXPERIMENTAL SYSTEM	FAR-TERM FULL-SCALE SYSTEM
IIA	FAR-TERM FULL-SCALE SYSTEM	

TABLE II: INTERLEAVING OF SYSTEM IMPLEMENTATION AND DESIGN TASKS FOR THE DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT

2. Each phase ends with a viable product. At the end of each phase the DMA has a working system for their continued evaluation.

3. Decision points are included at which time DMA can evaluate the progress to date, provide direction, and authorize continuation of work.

4. Phase I has been carefully planned to provide the maximum benefits from the resources required. The principal benefits of Phase I are: (a) Only one VAX system and software tool set will be procured, yet tool support will be available for all software lifecycle phases within 15 calendar months of contract start; (b) a link with the production mainframe computer will be accomplished to early establish MPE compatibility with the mainframe; (c) a total of 21 calendar months will be available for training, methodology development and evaluation, and (d) designs for Phase IA and Phase II will be completed to provide DMA early insight to the full MPE implementation.

The total cost for the implementation of the DMA MPE is estimated to be $11 million. This estimate includes the procurement of 9 VAX-11/780 computers and associated terminals and hardware, 7 software tool sets, maintenance for the hardware and software procurements, and contractor labor.

The hardware procurements cost approximately $4.2 million; software procurements cost approximately $2.5 million, and contractor labor costs approximately $4.3 million. The cost estimates by phase, funding type, and fiscal year are shown in Table III.

PHASE	FUNDING TYPE	COSTS (in Thousands of Dollars)						TOTALS BY PHASE
		FY 83	FY84	FY 85	FY 86	FY 87	FY 88	
I Near-Term Experimental System	R&D	90	1,210	690				1,990
IA Near-Term Full-Scale System	Production			1,490	3,630	120		5,240
II Far-Term Experimental System	R&D				770	730		1,500
IIA Far-Term Full-Scale System	Production					1,670	580	2,250
TOTALS BY FUNDING TYPE	R&D	90	1,210	690	770	730	–	3,490
	Production	–	–	1,490	3,630	1,790	580	7,490
TOTAL BY FISCAL YEAR	TOTAL	90	1,210	2,180	4,400	2,520	580	10,980

TABLE III: COST ESTIMATES FOR IMPLEMENTATION OF DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT

An estimate of the savings realized by using the MPE as compared to continued use of existing DMA methods of software development and maintenance was also calculated. The inputs to the savings estimate include productivity improvements due to the software tools, percentage of DMA activity in each lifecycle phase, the DMA programming population, an estimate for percentage savings as a function of time, and the DMA workyear cost including inflation. The cumulative net savings (cumulative net savings = sum of (yearly savings - yearly costs)) due to the DMA MPE capabilities is shown in Figure IV.

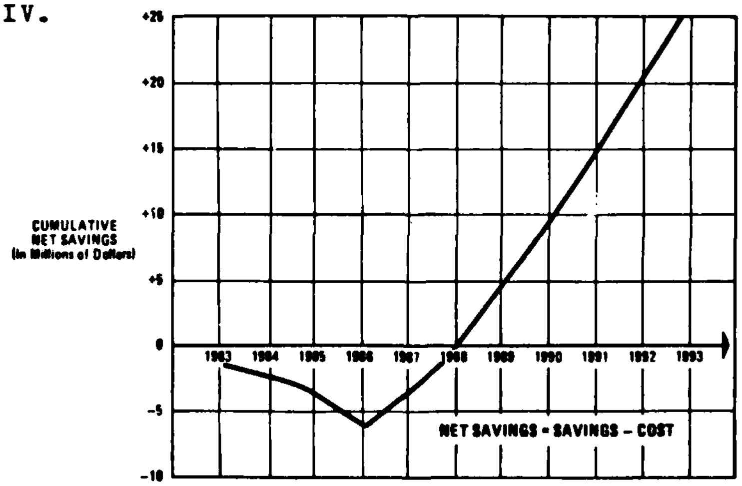


Figure IV: CUMULATIVE NET SAVINGS REALIZED WITH THE DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT

The costs of the DMA MPE will be recovered after five years (in 1988), and after ten years (in 1993) an estimated cumulative net savings of $25 million will be realized. The DMA Modern Programming Environment is definitely cost effective.

IV. Summary of General Dynamics' Technical Approach

General Dynamics Data Systems Division (GD/DSD) accomplished the Defense Mapping Agency Modern Programming Environment study in 21 months (from January 1981 to September 1982) using a four stage approach. The study milestone schedule and a block diagram of the stages of the technical approach are shown in Figure V.

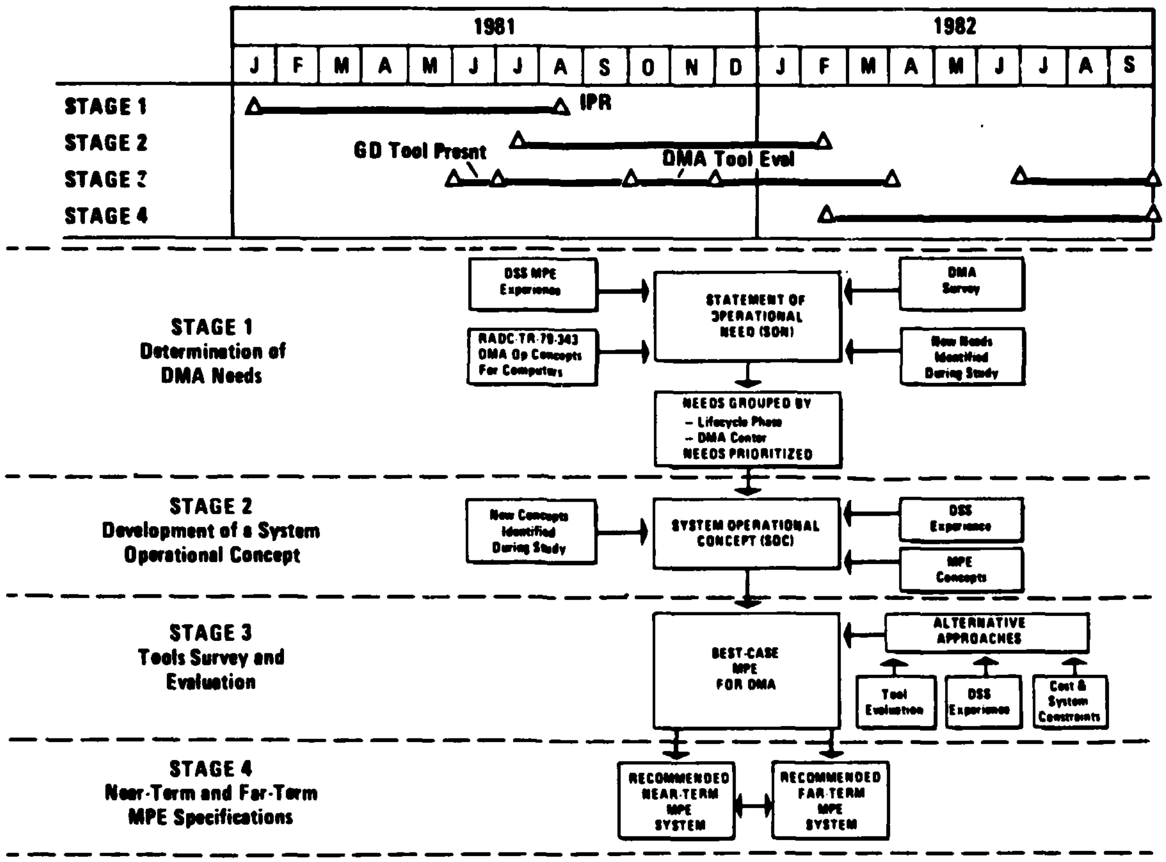


Figure V: GENERAL DYNAMICS' TECHNICAL APPROACH TO THE DEFENSE MAPPING AGENCY MODERN PROGRAMMING ENVIRONMENT STUDY

The objective of the first stage of the study was to identify the Defense Mapping Agency needs for a modern programming environment. General Dynamics distributed a five-part questionnaire to management and technical personnel at DMAAC, DMAHTC, and DMAHQ to ascertain the basic data for identification of DMA needs. A total of 181 questionnaires were returned. Personal interviews with DMA representatives were then conducted to gain additional insights into DMA needs. Stage 1 concluded with a list of 40 generic needs,

categorized by software lifecycle phase, and weighted by importance.

In Stage 2 of the study, 23 system operational concepts were formulated that satisfied the identified needs. A matrix of needs vs. concepts was created to specify which needs were satisfied by which concepts.

In Stage 3, 25 different software tools were presented and demonstrated at DMAAC and DMAHTC during a two week period at each center. The DMA comments from these presentations were analyzed, and 8 tools that supported all software lifecycle phases were selected for an in-depth, 8 week DMA evaluation at each center. The test-bed used for the evaluation process was the digital land mass problem. Once again, the DMA comments were collected and analyzed. Subsequently, the USE.IT software tool was of particular interest to the MPE study participants because of its requirements definition, design, and automated FORTRAN coding capabilities. Hence, it was decided to evaluate the applicability of USE.IT to the DMA software environment by solving a realistic DMA problem. The chosen problem was a long range navigation (LORAN) lattice calculation, and the evaluation was conducted from July to September, 1982. During this period the LORAN problem was modeled with USE.IT, executable code was produced, and the graphics displays were demonstrated to DMA. Stage 3 ended with a "best-case" modern programming environment model. This best-case was formulated by rating all available tools that satisified the needs and concepts identified in Stages 1 and 2 and then selecting those tools that best satisfied these needs and concepts. The rating was accomplished by using the concept implementation evaluation sheets to ensure traceability to needs and concepts, proper weighting of evaluation criteria, and consistency. A total of 173 concept implementation evaluation sheets were completed.

Finally, in Stage 4 this best-case modern program environment model was modified to satisfy the objectives and constraints of the near-term and far-term DMA modern programming environment. Typical constraints included cost, maturity of tools, availability of tools on the tool bearing host, continued support of DMA FORTRAN and COBOL efforts, user-friendliness of tools, vendor support, logical integration of tools to support the entire lifecycle phase, and smooth transition from the near-term to the far-term configuration. An additional consideration was the impact of the DMA Software Improvement Program (SIP) upon the specification of the modern programming environment. The objectives of the SIP were identified and found to support the system

operational concepts of the MPE. Therefore, compatibility of the MPE with SIP was a goal in the formulation of the near-term and far-term DMA modern programming environment configurations. Stage 4 culminated in the specification of the Defense Mapping Agency Modern Programming Environment configuration as shown in Figure I and the software tool list as shown in Table I.

A vitally important constituent of the General Dynamics' technical approach was the continual interaction among General Dynamics, DMA, and RADC. General Dynamics spent 92 workdays (during 63 calendar days) on-site at DMAHTC and DMAAC for technical interchange and data gathering. An additional 14 separate trips were made to DMAHTC, DMAAC, DMAHQ, and RADC for status reviews, oral presentations, and documentation preparation. Telephone communications among General Dynamics, DMA, and RADC were extensively used to keep all team members abreast of the Modern Programming Environment project status. These activities ensure that our DMA Modern Programming Environment specification will satisfy DMA needs and will be compatible with future DMA plans.

V. Conclusions and Recommendations

The fundamental conclusion of this Interactive Computer Program Development System Study is that the Modern Programming Environment specified in this document satisfies known Defense Mapping Agency requirements for the introduction of state-of-the-art software engineering technology into DMA's operational procedures. In addition, the plan for the implementation of this Modern Programming Environment is orderly, well-structured, and cost effective.

The major benefits to the DMA of this Modern Programming Environment specification are:

1. The technical hardware/software configuration is flexible and can easily grow and adapt to future DMA needs.

2. The VAX based MPE provides the technology base for rapid realization of productivity improvement in both software development and maintenance.

3. The MPE will pay for itself in five years and continue to accumulate net savings every year thereafter.

4. The introduction of the MPE will not disrupt DMA's production operations.

A modern programming environment is more than software tools hosted on a computer system. The tool bearing host computer and the complement of software tools shown in Figure I and Table I form the core of the MPE. It is recommended that the following items be considered as part of the total scope of the DMA Modern Programming Environment:

1. Management directives and commitments to the development and support of the MPE are required to ensure continuity of the MPE across all DMA software development and maintenance efforts. In particular, software contractors need DMA management direction to use development techniques and tools that enable easy DMA maintenance of the delivered software using the MPE.

2. The establishment of standards, guidelines, reviews, and methodologies for software

development and maintenance are needed. The work begun in these areas by the Software Improvement Program is the correct first step. There needs to be a continuation of the already existing co-ordination between the Software Improvement Program and the Modern Programming Environment study/implementation.

3. Personnel training in the proper and efficient use of the software tools is vital to realize the estimated productivity improvements. Training is a short-term cost with many-fold, long-term benefits.

Finally, as the result of our study, General Dynamics recommends the Defense Mapping Agency proceed with the implementation of the Modern Programming Environment specified in this document.

1.0 INTRODUCTION

General Dynamics Data Systems Division (GD/DSD) was contracted to perform an Interactive Computer Program Development System Study (ICPDSS) for the Defense Mapping Agency (DMA) under contract to Rome Air Development Center (RADC). As a result of this study GD/DSD has developed a complete design specification for a modern programming environment (MPE) for use by DMA by 1987. The technical approach to the study was organized into four distinct stages:

1) Determination of Defense Mapping Agency needs
2) Formulation of system concepts to satisfy those needs
3) Creation of the best-case model for a modern programming environment
4) Application of constraints to this model to arrive at near-term (1985) and far-term (1987) system recommendations.

This process is illustrated in Figure 1.1. In the implementation of this process close co-ordination and communication was planned and established with each of the two DMA centers.

Sections 1 and 2 of the report describe Stage one of the technical approach, and the development of the DMA Statement of Operational Need (SON) (Figure 1.2). The sources of information for the SON were government documents dealing with previous DMA studies, a GD/DSD survey questionnaire, and personnel interviews conducted by GD/DSD project team members at the DMA Hydrographic/Topographic Center (DMAHTC) and the DMA Aerospace Center (DMAAC). The results of the questionnaire were evaluated using a database inquiry system. These results along with an additional list of needs derived from government documents were used in formulating the original SON. The SON lists DMA needs and rates them on a 1 to 5 scale where 1 implies a low need and 5 a high need. The columns present the ratings as determined by each center and by General Dynamics. The process used is described in detail in Section 2.0 of this report. In June 1981, meetings were held at each center to validate the original SON findings. These meetings resulted in revisions to the SON and are described fully in Section 3.0.

Stage two, described in Sections 4 through 7, involved the development of a SON/SOC matrix providing a mapping of the operation needs identified in the SON into one or more generic programming concepts which satisfy each need. A

complete explanation of the development of this matrix and how to read and use it is found in Section 4.0. On 18-19 August 1981 the SON/SOC matrix was presented at an In-Process-Review (IPR) at RADC. As a result of this review several changes were made to the SON and the SOC's. The changes, described in Section 5.0, have been incorporated and the current SON and SON/SOC matrix are what appears in Figures 1.2 and 1.3.

Figure 1.1

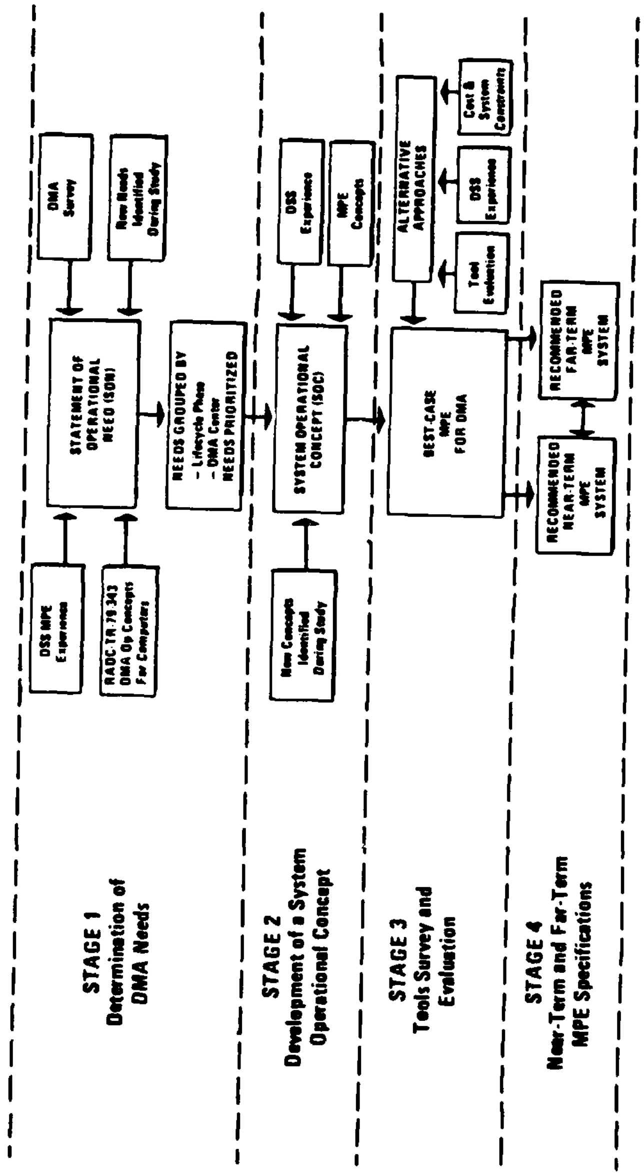


Figure 1.1 Technical Approach

09 OCTOBER 1981

```
SSSSSSSS  OOOOOOOO  NN    NN
SSSSSSSS  OOOOOOOO  NNN   NN
SS    SS  OO    OO  NNN   NN
SS        OO    OO  NNNN  NN
SS        OO    OO  NNNN  NN
SSSSSSSS  OO    OO  NN NN NN
SSSSSSSS  OO    OO  NN NN NN
      SS  OO    OO  NN  NNNN
      SS  OO    OO  NN  NNNN
SS    SS  OO    OO  NN   NNN
SSSSSSSS  OOOOOOOO  NN   NNN
SSSSSSSS  OOOOOOOO  NN    NN
```

	BASIC RATING		
CATEGORY	DMAAC	DSD	DMAHTC
PROJECT MANAGEMENT			
10 IMPROVED MILESTONE IDENTIFICATION	3	3	5
12 IMPROVE MANLOADING	4	4	4
14 IMPROVE SCHEDULE IMPACT ANALYSIS	3	4	5
48 CHARGEBACK SYSTEM	3	5	3
56 MANAGEMENT TRACKING FUNCTIONS	3	5	3
REQUIREMENTS			
1 FORMAL REQUIREMENTS SPECIFICATION	4	3	5
5 REQUIREMENTS TRACKING	3	3	4
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
DESIGN			
21 SIMULATOR FOR DESIGN	3	3	4
22 PROGRAM DESIGN LANGUAGE	4	4	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
CODING			
55 MODERN SOURCE DATA ENTRY TECHNIQUES	5	5	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
TEST			
2 QA PROCEDURES AND GUIDELINES	3	5	5
21 SIMULATOR FOR DESIGN	3	3	4
36 GRAPHICS AIDS	5	4	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
MAINTENANCE			
9 CONFIGURATION CONTROL	5	5	5
40 HISTORICAL DATA BASE TECHNIQUES	3	3	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
58 PRODUCTION PROGRAM OPTIMIZATION	3	4	3
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
OTHER			
3 INTERACTIVE SYSTEM ACCESS	5	5	5
4 INCREASED NUMBER OF TERMINALS	5	5	5
11 DECREASED PAPERWORK	3	5	5
16 UPDATE OF OLD DOCUMENTATION	5	3	5
18 FASTER INTEGRATION OF NEW EMPLOYEES	3	3	5
34 AUTOMATED TEXT MANAGEMENT TOOL	3	3	5
41 ORGANIZATION TOOLS/TECHNIQUES INTERFACE	3	4	5
42 USER ASSISTANCE FUNCTION	3	4	3
44 ERROR RATE STANDARDS	3	3	3
46 REDUCE ACCOUNTING DATA REPORT ANOMALIES	3	3	3
47 COMPREHENSIVE TRAINING PROGRAM	5	5	5
52 DECREASE TURNAROUND TIME TO MINUTES	3	3	3
54 NATURAL LANGUAGE USER/SYSTEM INTERFACE	3	3	3
60 STANDARDIZED DEVELOPMENT HARDWARE	4	5	4

Figure 1.2   Statement of Operation Need (SON)

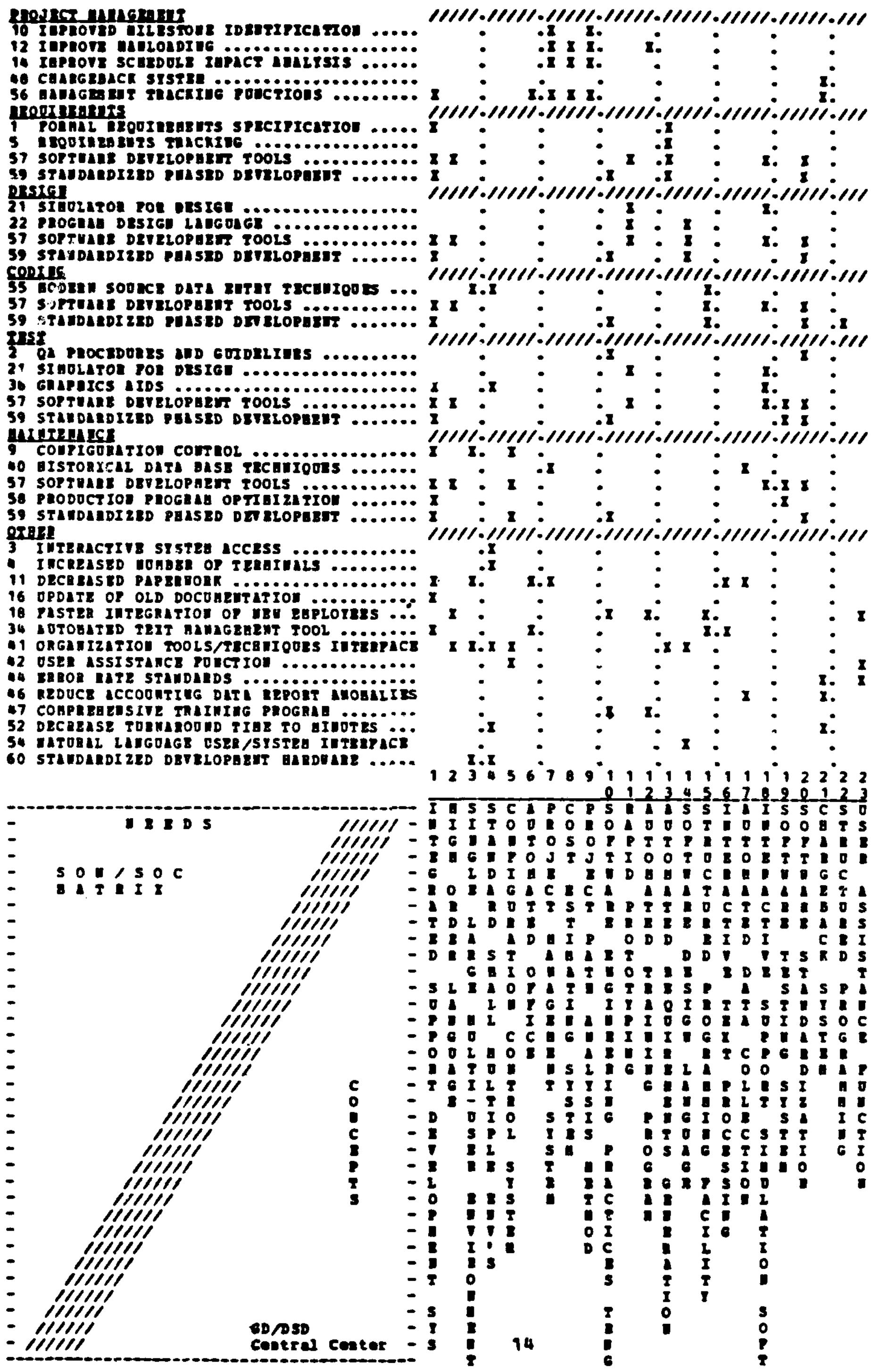


Figure 1.3 System Operational Concept (SOC)

A very important step in Stage 3 of this Interactive Computer Program Development System Study was the evaluation and selection of software tools for the DMA MPE. The software tools of interest in this study provide the capabilities and functions as outlined in the following paragraphs.

Automated software development tools serve as aids in the support of the software life cycle requirements, design, programming, testing and maintenance phases. These tools are provided to assist the manager, designer and programmer by automating part of the development process. Automation not only increases productivity, but it improves reliability and quality by using sound, well tested procedures with each program developed. Automated software tools provide an effective way to implement standards and conventions, and it improves the opportunity to reuse software and to reduce development costs. A general overview discussion of automated software tools is provided for information and insight into what is currently available and what is possible for future extensions.

Requirements tools allow a user to document and, in some implementations, analyze requirements in a succinct and unambiguous form. When analysis is possible a data base is constructed which is examined for consistency, completeness and traceability. USE.IT is such a tool. A requirements specification data base is built using a prompted, interactive interface language called AXES. Part of the output of USE.IT is a set of documents containing graphic and textual descriptions of the requirements of a specified software system in a format consistent with any other software system modeled using the tool. This documentation can then be used as input into the design phase of the software life cycle.

Design tools allow the user to document a design and perform an analysis to determine if it is technically viable. Both processes are only partially automated except in extremely narrow applications. SDDL is a design tool which performs these functions. A language based system, SDDL, documents the design in a concise structured syntax which is used to perform a small, but high level analysis. The information provided by the analysis, however, greatly decreases the effort required to manually evaluate the design's technical merit. A manual conversion of data would be required to convert the USE.IT output into a format acceptable to SDDL. Once complete, the design is manually translated into a computer program by use of a specified language. If the target language is known prior to the design phase (e.g.,

FORTRAN or COBOL) SDDL can be utilized in a manner that decreases the effort needed in the translation process.

Once translated into a computer program a software system must then be processed by language packages which are tools that transform the system into a state understandable by a digital computer. Compilers represent a major tool category within the realm of language packages. A compiler ties a specific implementation of a high level language to a specific computer architecture (i.e., the operations executed by the hardware).

Testing tools are used to evaluate the quality of software and demonstrate that it fulfills the needs documented in the user specifications. Testing tool capabilities include, but are not limited to, static analysis (performed on code including checks for program structure, complexity, and format), dynamic analysis (performed during program execution includes coverage analysis and assertion checking), automated test data generation, and output comparators.

Maintenance is the life cycle phase when software is placed in operational use. Tools used during this phase assist programmers in repairing or modifying existing production software systems. Repair and modification are primarily redevelopment activities which can be accommodated for the most part by the planned reuse of most or all of those development tools used during the requirements, design, coding, and testing phases. Additional maintenance tools include configuration control tools that are used to control changes to a production system and its documentation once it has been baselined. Configuration control tools maintain the current status program and it's documentation along with the past history of all code and documentation generated and changed.

The previously described software life cycle presupposes a defined problem exists which is known to be solvable. A corollary of this fact is that a problem definition step actually exists. Problem definition always occurs prior to requirement specification, however, analysis of solvability rarely occurs. One reason behind this fact is the labor intensive characteristic of feasibility studies. USE.IT also automates this process. Additional output of this tool not previously mentioned allows the specification of a system in a very high level language which is automatically translated into an executable form. In this manner a rapid prototype of a system may be generated and studied for technical viability.

As mentioned previously, part of the work to be accomplished in the ICPDSS was the evaluation and selection of software development tools to be included in the DMA near-term and far-term MPE's. This tool selection survey was divided into two phases. The goal of Phase I was the selection, demonstration and evaluation of a large number of software tools that are applicable to the DMA environment. The goal of Phase II was an in-depth analysis of tool capabilities for the DMA MPE using a DMA scenario as a test-bed problem. The objective was not to solve the test problem, but to determine metrics for tool comparisons.

In Phase I, a tool survey was conducted with information collected from a number of commercial and industrial sources and analyzed with respect to the DMA programming environment. Presentations and demonstrations were then conducted at each center on a set of tools covering all aspects of the software development process during the month of June 1981. The first two weeks of June were designated for presentations at DMAHTC with duplicate presentations at DMAAC the following two weeks. There were only minor differences between the sets of presentations given at the centers, relating to scheduling and not to material content. Figure 1.4 shows the tool presentation and demonstration schedule for each DMA center. Those tools outlined in cross-hatching were presented by their vendor; all other tools were presented by GD/DSD personnel. After each presentation a survey form was completed by the DMA attendees to evaluate the tool with respect to applicability and appropriateness to DMA needs. A copy of this survey form is included as Appendix B.

The tools were then ranked according to perceived need and applicability after the presentations by compiling statistics from the survey form. The tool rankings are presented in Figure 1.5 which is explained in the following three paragraphs.

The left-most column is the ranking of the tools based upon the number of survey responses with respect to their applicability to DMA tasks and ease of use in an interactive environment. The higher a tool appears in the list the larger the number of positive comments received. The capabilities of some demonstrated tools may have been new to the evaluators, and hence they did not immediately perceive a use for that tool. Therefore, in order to provide a uniform baseline for ranking familiar and unfamiliar tool capabilities, the DMA survey responses were also ranked by the number of least negative comments recorded. The middle column lists in order the tools based upon the number of <u>negative</u> survey responses. Again, the best tool is at the

top. The lower a tool appears in the list the larger the number of negative comments received. A line was drawn connecting the same tool in the most positive and least negative rankings. The more horizontal the slope of this line the higher the correlation between the two ranking schemes. A line with a very steep slope indicates an uncertain correlation. The most desirable tools are those that are near the top of both lists and have a high correlation indicated by a nearly horizontal line. Finally, the right-most column of Figure 1.5 is an independent assessment by the GD/DSD team personnel based upon the most positive ranking scheme. These rankings were utilized in the selection of tools for Phase II.

	MON	TUE	WED	THU	FRI	
at DMAHTC	1 AUTOFLOW UPDATE	2 FORMAT SFTRAN3 LOGOS	3 UIFOLA DAVE FTRAN'77 ANA	4 PRICE "S"*	5 STAR-1100*	9:30
	SDDL SCMS		////// /FAME/ //////	ATA COMPARATOR NODAL	/////// /LARE// ///////	1:00
	8 ///////// /OLIVER// /OSCAR/// /////////	9 //////////// /SOFTOOL 80/ ////////////	10/////////// /IS/1////// /WORKBENCH/ /FOR VAX/// ///////////	11 /////// /SOLID/ ///////	12///////// /1100//// /INTER-// /ACTIVE// /////////	9:30
	///////// /OLIVER// /OSCAR///	////////////// /SOFTOOL 80/// //////////////	///// /CUE/ /////		///////// /MAPPER// /////////	1:00
at DMAAC	15 ATA COMPARATOR NODAL	16//////// /1100/// /INTER-/ /ACTIVE/ ////////	17 UIFOLA DAVE FTRAN'77 ANA	18 /////// /SOLID/ ///////	19 STAR-1100	9:30
	SDDL SCMS	//////// /MAPPER/ ////////	///// /CUE/ /////		////// /LARE/ //////	1:00
	22 //////// /OLIVER/ ////////	23 //////////// /SOFTOOL 80/ ////////////	24/////////// /IS/1////// /WORKBENCH/ /FOR VAX/// ///////////	25 AUTOFLOW UPDATE	26 ////// /FAME/ //////	9:30
	/////// /OSCAR/ ///////	////////////// /SOFTOOL 80/// //////////////	FORMAT SFTRAN3 LOGOS	PRICE "S"		1:00

Figure 1.4 Presentation/Demonstration Schedule

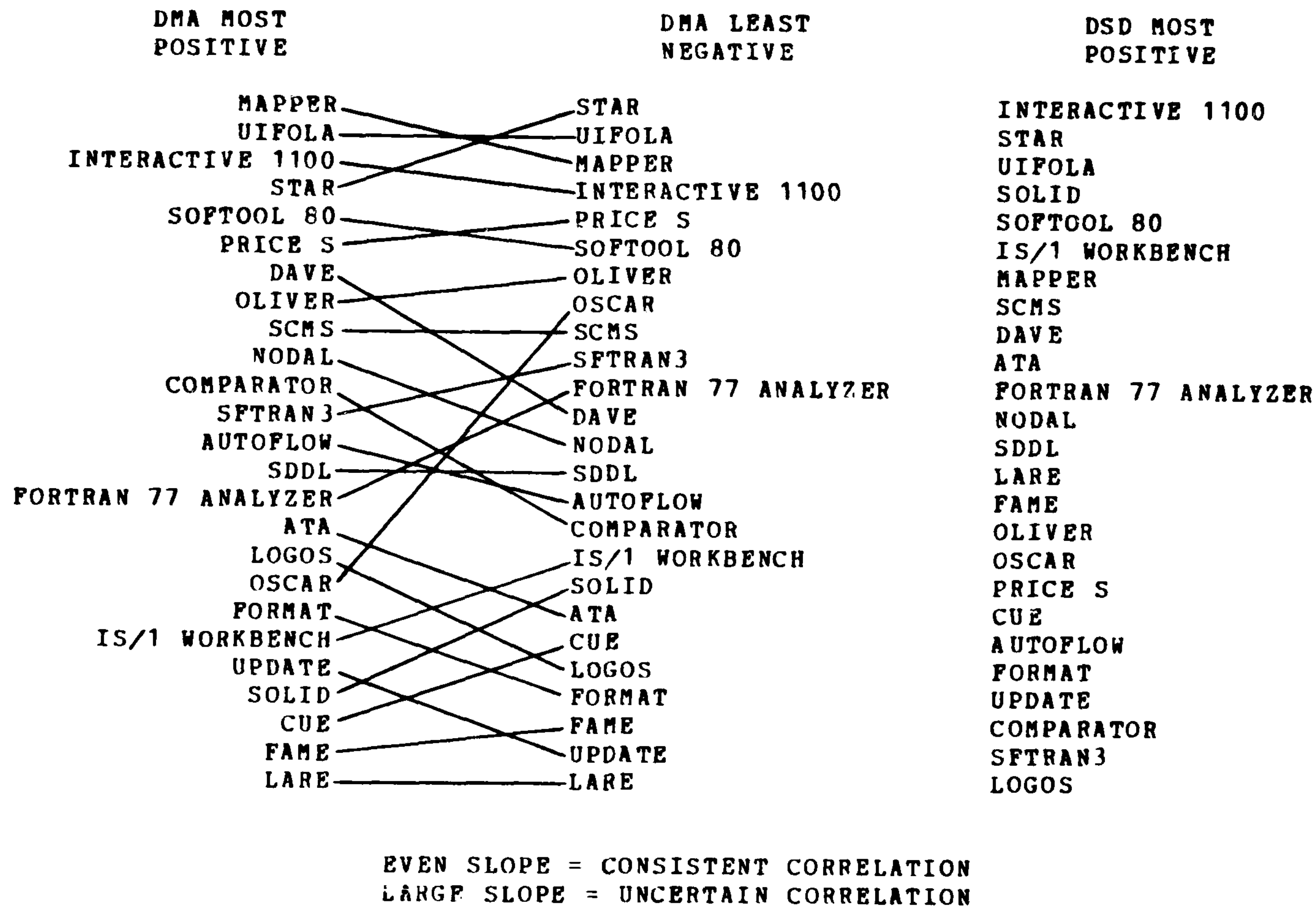


Figure 1.5 Tool Ranking from Demonstrations

In Phase II of the tool evaluation plan, a DMA software scenario was simulated using a test-bed problem that exercised the tool capabilities. The test-bed problem utilized digitized feature data and a simulated cartographic data base to extract a new manuscript. The software tools were used for requirements definition, design, coding and testing. In addition, a data base management tool was used in the collection of data from DMA team members for statistical evaluation; and a prepared tutorial on a project management tool was available for viewing. Figure 1.6 illustrates the schedule of activities as they occurred in Phase II. Subsequently, the USE.IT software tool was of particular interest to the MPE study participants because of its requirements definition, design, and automated FORTRAN coding capabilities. Hence, it was decided to evaluate the applicability of USE.IT to the DMA software environment by solving a realistic DMA problem. The chosen problem was a long range navigation (LORAN) lattice calculation, and the evaluation was conducted from July to September, 1982. During this period the LORAN problem was modeled with USE.IT, executable code was produced, and the graphics displays were demonstrated to DMA. A description of the LORAN problem is included as Appendix I.

Utilizing the data collected in Phase I and Phase II, Near-Term and Far-Term MPE's were developed. These recommended near-term and far-term environments meet the requirements as specified in the SON/SOC as well as provide for the environmental capabilities identified during the software tool evaluation. In the Near-Term MPE risks have been minimized by recommending tools which are currently available and have been throughly investigated with respect to claimed performance capabilities. Performance cannot be quantified, but cost data and rationale are provided which support our conclusions. An experimental system would be developed first in the implementation of the environment to provide engineering data to fine tune system performance. Further information on the experimental system can be found in Section 19.1.

```
.(--SEP--)(--------------------------OCTOBER-------------------------).
.M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS.
.28 29 30 1  2     5  6  7  8  9     12 13 14 15 16    19 20 21 22 23   .
.SSSSSS            WWWWWWWWWWWW      SSSSSSSSSSSS                SSSSSS .R. BOND
.(STATUS           WWWWWWWWWWWW                                  WWWWWW .M. GOODE
. REVIEW)          #----------#      #----------#                #----# .
.                                                             HHHBBBBBB .PROJECT MANAGEMENT
.                  HHH               AAA                                .INTRODUCTION
.                    HHHHHHH           AAAAAAA                          .REQUIREMENTS
.                         HHHHHH     HHHHHHHHHBBB    AAAAAAAAAAAAAAA    .DESIGN
.                                                    HHHHHHHHHHHHHHH    .CODING
.                                                                       .
.(---OCTOBER----)(----------------------NOVEMBER---------------------).
.M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS.
.26 27 28 29 30    2  3  4  5  6     9  10 11 12 13    16 17 18 19 20   .
.                  SSSSSSSSS                           WWWWWWSSSSSSSS   .R. BOND
.      SSS         WWWWW:WWW       SSSWWW              WWWWWWSSSSSSSS   .M. GOODE
.                  #-------#                           #------------#   .
.                                                                       .DESIGN
.BBBBBBBBBBBBBBB   BBBBBB                                               .CODING
.                       BBBBBBBBBB     BBBB   BBBBBB                    .TESTING
.                                                      HHHHHHHAAAAAAAA  .END AND CLOSE
.-----------------------------------------------------------------------.

WWWWWWW - AT WASHINGTON D.C.
SSSSSSS - AT ST. LOUIS, MO
#-----# - GD/DSD ON SITE SUPPORT*
HHHHHHH - DMAHTC WORK
AAAAAAA - DMAAC WORK
BBBBBBB - SAME PHASE IN WORK AT BOTH CENTERS
```

Figure 1.6 Tool Evaluation Schedule of Activities
(page 1 of 2)

```
.(_________JULY_________)(________AUGUST________)(______SEPTEMBER_______).
.________________________________1982___________________________________.
.                                                                       .
.                                 #__#                                  .INITIAL TRIP BY GD
.                                                                       . TO HOS
.                                                                       .
.                                  #___________________________________#.DEVELOPMENT OF LORAN
.                                                                       . SOLUTION WITH USE.IT
.                                                                       .
.                                                  #__#                 .GD 5 DMA AT HOS
.                                                                       .
.                                                                    #  .VIDEOTAPE OF LORAN
.                                                                       . DEMONSTRATION/PRESENTATION
.                                                                       . DELIVERED
.                                                                       .
.                                                                      #.FORTRAN SOURCE TAPE
.                                                                       . OF LORAN PROBLEM
.                                                                       . DELIVERED TO
.                                                                       . DMAAC & DMAHTC
.                                                                       .
.-----------------------------------------------------------------------.
```

Figure 1.6 Tool Evaluation Schedule of Activities
(page 2 of 2)

The near-term and far-term recommendations are summarized in Figures 1.7 and 1.8 respectively. The Near-Term MPE is based upon a VAX configuration. This configuration provides a software development capability with minimum schedule and technical risk at low cost. These systems represent the state-of-the-art in software development tools when constrained by DMA's current systems and future plans. The Far-Term MPE is also based upon the VAX because of the abundance of software tools currently available and projected to be available for this system.

The effort involved in the development of the report and its associated annexes included over three manyears of labor by GD/DSD with 92 mandays (63 calendar days) of activity being conducted on-site at the DMA centers.

The acquisition of the Near-Term MPE tools and tool bearing host (TBH) and its evolution to the Far-Term MPE will not satisfy all the software development support requirements for the 1987 target date. The support areas of cost estimating, management tools, tool set integration, code auditors, and Ada will require research and development activities. In each area work must be accomplished to define the DMA specific needs, identify solutions, and provide for the solutions to be integrated into the Far-Term MPE.

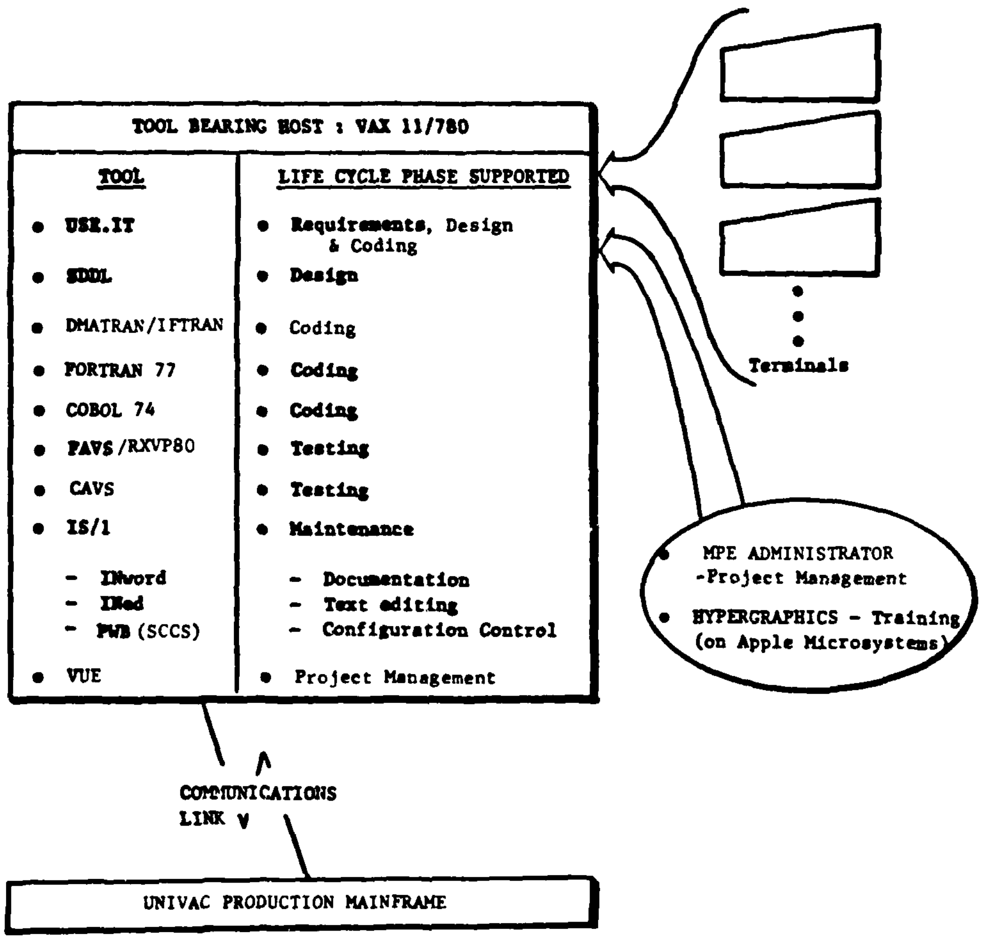


Figure 1.7 Near-Term System Configuration for DMA Mo, ern Programming Environment

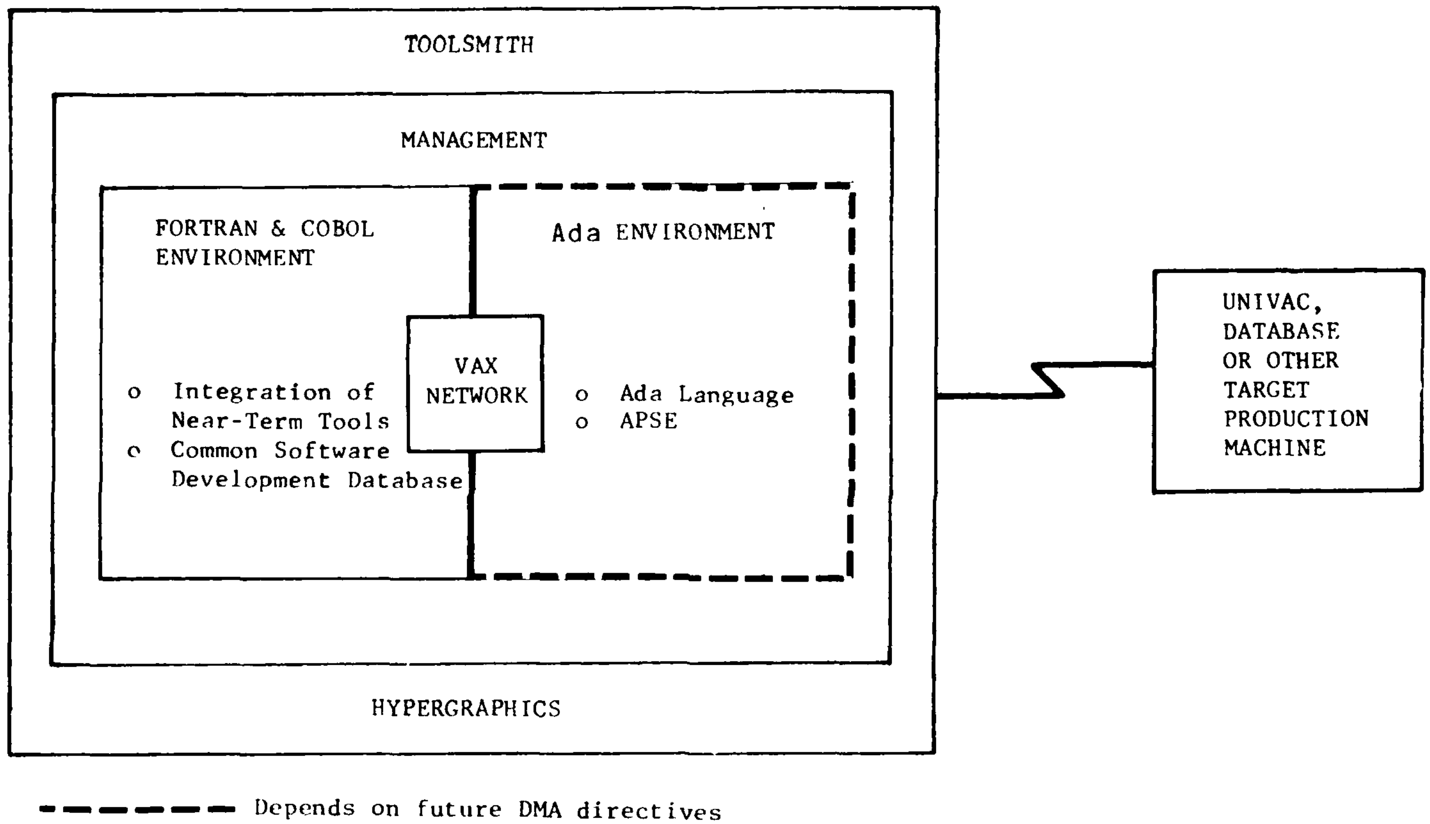


Figure 1.8 Far-Term System Configuration for DMA Modern Programming Environment

2.0 DETERMINATION OF DMA NEEDS

The first stage of this MPE study as stated in the technical approach was the determination of DMA needs. This included the needs common to both DMA centers and those specific to DMAHTC and DMAAC. The following list of sources was used to obtain data which in turn was analyzed and the results presented in the SON described in Section 2.3.

1) FEDSIM (Federal Computer Performance Evaluation and Simulation Center) Installation Review - DMAHTC - November 1980
2) DMA Operational Concepts (1982 - 1990) - May 1979
3) DMA Programming Support Library (PSL) Interim Evaluation Report, IBM/FSD - November 1980
4) DMAAC/Scientific Computer Division - Software Life Cycle Standards - February 1981
5) DMAAC Organizational Mission Functions - October 1980
6) FEDSIM Installation Review - DMAAC - August 1980
7) DMA Modern Programming Environment (MPE) - January 1980
8) FEDSIM Optimization and Error Rate Studies - February 1981
9) Operational Improvement Opportunities for UNIVAC 1100/80
10) DMAHTC Organizational Manual
11) General Dynamics DMA Survey - March 1981
12) Interviews conducted by General Dynamics at DMAHTC and DMAAC
13) DMAAC Modern Programming Environment Pilot Project Evaluation Report
14) The DMAHTC Modern Programming Environment (MPE) Pilot Project

2.1 DMA SURVEY QUESTIONNAIRE

Item 11 of the above list (see Appendix A) consisted of a questionnaire developed by General Dynamics to help determine needs at DMA and help identify currently used tools appropriate for common use. The questionnaire was also planned to function as a tool in validating the findings of the Boeing Report, RADC-TR-79-343 (item 7, DMA MPE - January 1980), as well as a means of gathering information about the future plans of DMA in the areas of operations and policies. The questionnaire corroborated the findings of the Boeing Report with minor exceptions in the area of project management techniques. Since the Boeing Report was generated, DMA has started activities to correct identified deficiencies.

The questionnaire consisted of five parts. The first and last sections were to be answered by every respondent. The first, the "respondent" section, was used to correlate

answers with respect to a person's background. This also included questions to determine DMA organization (DMAAC vs DMAHTC), environment (open vs closed shop) and security (Sensitive Compartmented Information (SCI) vs collateral) which were to be used in the classification of needs. The last, the tools section, was included to gather general knowledge about what software tools exist at DMA and their usefulness. One of each of the three remaining sections was to be answered by each respondent according to his job classification. These included a technical section to gather data on operations, a management section to determine methods of operation and a policies section to be answered by higher management concerning DMA planning, control, organization and direction.

230 questionnaires were distributed, 10 to DMAHQ, 110 to DMAHTC and 110 to DMAAC. 181 were completed and returned, 43% from DMAAC and 57% from DMAHTC. There were 28 invalid questionnaires (out of the 181) due to one of the following not being given: DMA organization (DMAAC vs DMAHTC), environment (open vs closed shop) or security (SCI vs collateral). No attempt was made to validate these questionnaires with additional information because the valid sample size was considered sufficient.

Data from the DMA survey questionnaire was collected and stored in a database inquiry system to be used in compiling data for the SON.

2.2 DMA PERSONNEL INTERVIEWS

The personnel interviews, item 12, were conducted during March, April and May, 1981 with representatives at each center and were used to gain additional insights into DMA center activities and to gather supporting information on their needs.

2.3 STATEMENT OF OPERATION NEEDS (SON)

The resulting SON (Figure 2.1) has three major columns, BASIC DATA and NORMALIZED DATA by area (working environment) and a list of DMA needs.

The basic data represents actual responses to database inquiries (for numeric data) and annotations from manuals in which needs were presented (alphanumeric data). Note that needs identified in associated manuals were included only if a similar category was not present in the survey responses (numeric data).

The normalized data resulted from the fact that the number of responses in any particular area were different from the number of responses in the other areas. To form a common base the numeric basic data was normalized by forming percentages with the total number of responses addressing a particular need.

Within the basic and normalized columns the data was broken out additionally by common management, closed shop non-secure, closed shop secure, open shop non-secure and open shop secure using the respondent background information obtained in the first section of the questionnaire and according to the following definitions:

1) Common management represents common needs across all shops as perceived by management.

2) Secure vs non-secure is the same as SCI vs collateral.

3) Open shop meant the computer could be accessed by all qualified individuals.

4) Closed shop meant a restricted staff was assigned for computer use and operation.

```
SSSSSSSS  OOOOOOOO  NN    NN
SSSSSSSS  OOOOOOOO  NNN   NN
SS    SS  OO    OO  NNN   NN
SS        OO    OO  NNNN  NN
SS        OO    OO  NNNN  NN
SSSSSSSS  OO    OO  NN NN NN
SSSSSSSS  OO    OO  NN NN NN
      SS  OO    OO  NN  NNNN
      SS  OO    OO  NN  NNNN
SS    SS  OO    OO  NN   NNN
SSSSSSSS  OOOOOOOO  NN   NNN
SSSSSSSS  OOOOOOOO  NN    NN
```

A: FEDSIM REVIEW - DMAAC - AUG 1980
B: RADC-TR-79-343
E: FEDSIM ERROR RATE STUDY - FEB 1980
M: FEDSIM REVIEW - DMAHTC - NOV 1980
O: DMA OPERATIONAL CONCEPTS
P: PSL REPORT

	BASIC DATA										NORMALIZED DATA											
	DMAAC					DMAHTC					DMAAC					DMAHTC						
	COMMON MANAGEMENT	CLOSED SHOP NON SECURE	CLOSED SHOP SECURE	OPEN SHOP NON SECURE	OPEN SHOP SECURE	COMMON MANAGEMENT	CLOSED SHOP NON SECURE	CLOSED SHOP SECURE	OPEN SHOP NON SECURE	OPEN SHOP SECURE	COMMON MANAGEMENT	CLOSED SHOP NON SECURE	CLOSED SHOP SECURE	OPEN SHOP NON SECURE	OPEN SHOP SECURE	COMMON MANAGEMENT	CLOSED SHOP NON SECURE	CLOSED SHOP SECURE	OPEN SHOP NON SECURE	OPEN SHOP SECURE		CATEGORY
	-	3	1	11	6	-	5	1	9	8	-	16	8	30	25	-	29	11	24	31	1	FORMAL REQUIREMENTS SPECIFICATION
	5	1	4	14	8	10	4	3	12	13	36	5	31	38	33	36	24	33	32	50	2	QA PROCEDURES OR GUIDELINES
	8	6	1	11	5	15	6	5	12	9	57	32	8	30	21	54	35	56	32	35	3	INTERACTIVE SYSTEM ACCESS
	-	9	7	17	13	-	8	7	16	13	-	47	54	46	54	-	47	78	43	50	4	INCREASED NUMBER OF TERMINALS
	-	7	6	16	14	-	6	4	13	14	-	37	46	43	58	-	35	44	35	54	5	REQUIREMENTS TRACKING
	13	3	3	6	6	23	6	4	8	11	93	16	23	16	25	82	35	44	22	42	6	COMMON LANGUAGE
	-	7	2	11	5	-	4	3	13	12	-	37	15	30	21	-	24	33	35	46	7	COMMON HARDWARE
	-	8	7	10	14	-	5	7	7	15	-	42	54	27	58	-	29	78	19	58	8	MODULE INTERFACE DEFINITION
	-	8	5	10	11	-	6	8	10	17	-	42	38	27	46	-	35	89	27	65	9	CONFIGURATION CONTROL
	5	7	2	11	1	18	1	1	2	5	36	37	15	30	4	64	6	11	5	19	10	IMPROVE MILESTONE IDENTIFICATION
	4	5	3	12	5	12	3	4	7	11	29	26	23	32	21	43	18	44	19	42	11	DECREASE PAPERWORK
	5	0	0	0	0	9	2	0	2	1	36	0	0	0	0	32	12	0	5	4	12	IMPROVE MANLOADING
	3	-	-	-	-	6	-	-	-	-	21	-	-	-	-	21	-	-	-	-	13	IMPROVE COST ANALYSIS
	9	-	-	-	-	3	-	-	-	-	64	-	-	-	-	11	-	-	-	-	14	IMPROVE SCHEDULE IMPACT ANALYSIS
	9	1	3	3	7	5	3	5	7	8	29	5	23	8	29	18	18	56	19	31	15	MODERNIZATION
	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	7	-	-	-	-	16	UPDATE OF OLD DOCUMENTATION
	-	1	1	2	2	2	1	1	4	5	-	5	8	5	8	7	6	11	11	19	17	STANDARDIZATION
	2	-	-	-	-	5	-	-	-	-	14	-	-	-	-	18	-	-	-	-	18	FASTER INTEGRATION OF NEW EMPLOYEES
	2	7	4	14	9	16	7	5	9	8	11	37	31	38	38	43	41	56	24	31	19	REQUIREMENTS DEFINITION TOOL
	3	6	4	13	9	23	5	4	10	12	17	32	31	35	38	62	29	44	27	46	20	REQUIREMENTS VALIDATION TOOL
	4	5	2	11	6	16	6	7	12	14	22	26	15	30	25	43	35	78	32	54	21	SIMULATOR FOR DESIGN
	2	10	6	14	8	17	7	6	11	12	11	53	46	38	33	46	41	67	30	36	22	PROGRAM DESIGN LANGUAGE
	4	9	6	14	9	18	5	4	12	11	22	47	46	38	38	49	29	44	32	42	23	STANDARDIZATION TOOL
	0	1	1	2	2	3	2	1	2	3	0	5	8	5	8	8	12	11	5	12	24	COMPILER TOOL
	0	1	1	1	1	2	2	1	1	2	0	5	8	3	4	5	12	11	3	8	25	MAP TOOL
	0	1	1	1	1	2	2	1	1	3	0	5	8	3	4	5	12	11	3	12	26	ASSEMBLER TOOL
	0	0	0	1	0	3	2	3	2	4	0	0	0	3	0	8	12	33	5	15	27	LINKAGE EDITOR TOOL
	0	3	3	3	2	4	2	1	3	2	0	16	23	8	8	11	12	11	8	8	28	TEXT EDITOR TOOL
	1	3	3	5	3	6	4	5	6	8	6	16	23	14	13	16	24	56	16	31	29	CONFIGURATION CONTROL SYSTEM
	1	2	1	3	3	4	2	2	4	4	6	11	8	8	13	11	12	22	11	15	30	SECURITY SYSTEM
	5	11	8	16	10	17	10	6	20	17	28	58	62	43	42	46	59	67	54	65	31	TEST GENERATION TOOL
	5	10	7	16	10	15	9	7	19	17	28	53	54	43	42	41	53	78	51	65	32	TEST VALIDATION TOOL
	1	7	5	12	8	12	5	4	7	7	6	37	38	32	33	32	29	44	19	27	33	FLOWCHART TOOL
	2	3	1	8	4	13	3	6	13	14	11	16	8	22	17	35	18	67	35	54	34	AUTOMATED TEXT MANAGEMENT SYSTEM
	4	6	1	12	6	22	8	4	16	11	22	32	8	32	25	59	47	44	43	42	35	SOFTWARE DOCUMENTATION LANGUAGE
	4	7	3	11	5	26	9	6	16	15	22	37	23	30	21	70	53	67	43	58	36	GRAPHICS AIDS
	3	6	4	10	7	19	1	3	7	8	17	32	31	27	29	51	6	33	19	31	37	COST ESTIMATING TOOL
	2	6	3	10	6	18	1	3	4	6	11	32	23	27	25	49	6	33	11	23	38	BUDGET TRACKING TOOL
	2	7	4	12	6	17	3	4	9	10	11	37	31	32	25	46	18	44	24	38	39	REPORT GENERATOR TOOL
	3	5	3	11	5	18	1	4	7	9	17	26	23	30	21	49	6	44	19	35	40	HISTORICAL DATA BASE TOOL
	2	4	1	10	6	16	3	2	10	5	11	21	8	27	25	43	18	22	27	19	41	ORGANIZATIONAL TOOLS/TECHNIQUES INTER
			E					E					E					E			42	USER ASSISTANCE FUNCTION
			E A					E M					E A					E M			43	BETTER TAPE PROCEDURES
			E					E					E					E			44	ERROR RATE STANDARDS
			E										E								45	REPLACE/TERMINATE DCT 2000'S
			E					E					E					E			46	REDUCE ACCOUNTING DATA REPORT ANOMALI
			A B					E B					A B					M B			47	COMPREHENSIVE TRAINING PROGRAM
			A					M					A					M			48	CHARGEBACK SYSTEM
								M										M			49	SYSTEM CHANGE BULLETIN
			A B					B					A B					B			50	PROGRAMMING AND OPERATIONS STANDARDS
			O					O					O					O			51	REMOTE ACCESS BY FEDERAL AGENCIES
			O					O					O					O			52	DECREASE TURNAROUND TIME TO MINUTES
			O					O					O					O			53	MACRO ANALYTICAL/SIMULATION MODEL
			O					O					O					O			54	NATURAL LANGUAGE USER/SYSTEM INTERFAC
			O B					O B					O B					O B			55	MODERN SOURCE DATA ENTRY TECHNIQUES
			P					P					P					P			56	MANAGEMENT TRACKING FUNCTIONS
			P B					P B					P B					P B			57	SOFTWARE DEVELOPMENT TOOLS
			B					B					B					B			58	PRODUCTION PROGRAM OPTIMIZATION
			B					B					B					B			59	STANDARDIZED PHASED DEVELOPMENT

Figure 2.1 Original SON

3.0 SON VALIDATION

During the month of June, 1981 meetings were held at DMAHTC and DMAAC to validate the findings presented in the SON (Figure 2.1).

The first meeting was held at DMAHTC with General Dynamics Central Center (DSD/Central Center) project personnel and two DMAHTC techniques office representatives. At this meeting the SON was presented using the normalized data given in high, medium and low format, where high meant 68-99% of those responding saw a need, medium meant 34-67% and low meant 0-33%. Discussion topics were open vs closed shop, secure vs non-secure, continuation meetings with organizational representatives, and alternatives to the SON breakouts (open shop, closed shop, secure, non-secure). No consensus was reached due to ambiguities in the definitions of "access", "open shop" and "closed shop".

A second meeting at DMAHTC was attended by the DSD/Central Center project personnel and management representatives from multiple organizations. It was first decided that no concensus could be determined for definitions of open vs closed shop or for an alternative breakout (minicomputers vs mainframe for example). Therefore the breakouts were eliminated. Next the meanings of the needs and their applicability to DMA were defined. Several needs were discarded as they were covered by larger categories in the list. Finally all the needs identified were rated as high, medium or low needs.

A third meeting was held at DMAHTC with the DSD/Central Center project personnel and technical representatives from various organizations. Starting with the list of needs as set in the management meeting the needs were again defined and rated.

The SON meeting at DMAAC was attended by the DSD/Central Center project personnel and four DMAAC representatives from multiple organizations. The list of needs was presented as developed at DMAHTC and again defined and rated as high, medium or low needs.

Composites were made of the ratings by center and included inputs from DSD/Central Center project personnel (see Figures 3.1 and 3.2). The following rating scheme was used in the resulting revised SON (Figure 3.3):

5 = high
4 = medium high

3 = medium
2 = medium low
1 = low

The needs referenced in government documents, needs 42-59, were rated high if the need was called out in more than one document and medium if it was mentioned in only one. A blank in the rating column indicates that there was no need identified in that category for a particular DMA center. The extreme left-hand column of numbers serves to provide numeric reference and tracking to the original SON for each need; omissions in the sequence occur. For example, some were eliminated from the SON (see Figure 5.1) because they were covered by larger categories in the list (8,13,19,20,23,29,31,32,35,37,38,39,50), were too broad (15,17,30) or were outside the scope of the study (25,43,45,51,53).

Need Number(s)	Incorporated Into Need(s)
6,23	59
8	1,22,59
13	37
19,20	1
28	34
29	9
31,32	57
35	22
37,38,39	56
49	42

Need number 60 was added during DMA MPE study team meetings at DSD/Central Center in Fort Worth in July 1981 while refining the SON/SOC (System Operational Concept) matrix which will be discussed next; and was a redefinition of number 7. For needs 24,26,27 and 33 additional study determined there was no actual need currently or projected at DMA.

MEETINGS

NEED	MANAGEMENT	TECHNICAL	COMPOSITE
1	H	H	5
2	H	HM	5
3	H	H	5
4	H	H	5
5	H	M	4
6	M	M	3
7	M	H	4
9	H	HM	5
10	H	H	5
11		H	5
12	M	H	4
13	H	H	5
14	H	H	5
16	H	H	5
17	H	M	4
18	H	H	5
19	M	H	4
20	H	H	5
21	H	ML	4
22	H	HM	5
23	M	HM	4
24	L	L	1
26	L	L	1
27	M	L	2
28	L	M	2
29		H	5
30	M	M	3
31	H	M	4
32	H	H	5
33	L	ML	2
34	H	H	5
36	H	H	5
38	H	HM	5
39	M	M	3
40	H	HM	5
41	H	H	5

Figure 3.1 DMAHTC SON Data

REPRESENTATIVES

REPRESENTATIVES NEED	DSS	1	2	3	4	COMPOSITE
1	H	H	H	M	M	4
2	M		M	M	M	3
3	H	H	H	H	H	5
4	H	H	H	H	H	5
5	M		M	M	M	3
6	L	M	M	L	M	2
7	L	M	L	L	L	1
9	H	H	H	H	M	5
10	M	M	M	L	H	3
11	M	M	H	L	M	3
12	M	H	M	H	H	4
13	L	L	L	L	M	2
14	M	M	M	M	H	3
16	M-H	H	H	M	M	4
17	H	H	H	L	M	4
18	M	M	M	M	H	3
19	M	M	H	M	M	3
20	H	M	H	H	M	4
21	M	M	M	M	M	3
22	H	M	H	L	M	4
23	H	H	M	H	M	4
24	L	L	L	L	M	2
26	L	L	L	L	M	2
27	M	M	L	M	M	3
28	M	H	M	H	H	4
29	M	H	M	M	M	3
30	L	M	L	L	H	3
31	H	M	M	H	H	4
32	M	H	M	M	H	4
33	L	L	L	L	L	1
34	L	H	L	L	H	3
36	H	H	H	M	H	5
38	L	L	L	L	M	2
39	L	L	L	L	H	2
40	M	M	M	M	H	3
41	M	M	M	M	H	3

Figure 3.2 DMAAC SON Data

6 AUGUST 1981

```
SSSSSSSS  00000000  NN     NN
SSSSSSSS  00000000  NNN    NN
SS     SS 00     00 NNN    NN
SS        00     00 NNNN   NN
SS        00     00 NNNN   NN
SSSSSSSS  00     00 NN NN  NN
SSSSSSSS  00     00 NN NN  NN
       SS 00     00 NN   NNNN
       SS 00     00 NN   NNNN
SS     SS 00     00 NN    NNN
SSSSSSSS  00000000  NN    NNN
SSSSSSSS  00000000  NN     NN
```

CATEGORY	BASIC RATING DMAAC	BASIC RATING DMAHTC
1 FORMAL REQUIREMENTS SPECIFICATION	4	5
2 QA PROCEDURES OR GUIDELINES	3	5
3 INTERACTIVE SYSTEM ACCESS	5	5
4 INCREASED NUMBER OF TERMINALS	5	5
5 REQUIREMENTS TRACKING	3	4
6 COMMON LANGUAGE	2	3
9 CONFIGURATION CONTROL	5	5
10 IMPROVE MILESTONE IDENTIFICATION	3	5
11 DECREASE PAPERWORK	3	5
12 IMPROVE MANLOADING	4	4
13 IMPROVE COST ANALYSIS	2	5
14 IMPROVE SCHEDULE IMPACT ANALYSIS	3	5
16 UPDATE OF OLD DOCUMENTATION	4	5
18 FASTER INTEGRATION OF NEW EMPLOYEES	3	5
19 REQUIREMENTS DEFINITION TOOL	3	4
20 REQUIREMENTS VALIDATION TOOL	4	5
21 SIMULATOR FOR DESIGN	3	4
22 PROGRAM DESIGN LANGUAGE	4	5
23 STANDARDIZATION TOOL	4	4
24 COMPILER TOOL	2	1
26 ASSEMBLER TOOL	2	1
27 LINKAGE EDITOR TOOL	3	2
28 TEXT EDITOR TOOL	4	2
29 CONFIGURATION CONTROL SYSTEM	3	5
30 SECURITY SYSTEM	3	3
31 TEST GENERATION TOOL	4	4
32 TEST VALIDATION TOOL	4	5
33 FLOWCHART TOOL	1	2
34 AUTOMATED TEXT MANAGEMENT SYSTEM	3	5
36 GRAPHICS AIDS	5	5
38 BUDGET TRACKING TOOL	2	5
39 REPORT GENERATOR TOOL	2	3
40 HISTORICAL DATA BASE TOOL	3	5
41 ORGANIZATIONAL TOOLS/TECHNIQUES INTERFACE	3	5
42 USER ASSISTANCE FUNCTION	3	3
43 BETTER TAPE PROCEDURES	5	5
44 ERROR RATE STANDARDS	3	3
45 REPLACE DCT 2000'S	3	
46 REDUCE ACCOUNTING DATA REPORT ANOMALIES	3	3
47 COMPREHENSIVE TRAINING PROGRAM	5	5
48 CHARGEBACK SYSTEM	3	3
49 SYSTEM CHANGE BULLETIN		3
50 PROGRAMMING AND OPERATIONS STANDARDS	5	3
51 REMOTE ACCESS BY FEDERAL AGENCIES	3	3
52 DECREASE TURNAROUND TIME TO MINUTES	3	3
53 MACRO ANALYTICAL/SIMULATION MODEL	3	3
54 NATURAL LANGUAGE USER/SYSTEM INTERFACE	3	3
55 MODERN SOURCE DATA ENTRY TECHNIQUES	5	5
56 MANAGEMENT TRACKING FUNCTIONS	3	3
57 SOFTWARE DEVELOPMENT TOOLS	5	5
58 PRODUCTION PROGRAM OPTIMIZATION	3	3
59 STANDARDIZED PHASED DEVELOPMENT	3	3
60 STANDARDIZED HARDWARE	4	4

Figure 3.3 Revised SON

4.0 SYSTEM OPERATIONAL CONCEPTS (SOC)

The second stage of this modern programming environment study as stated in the technical approach was the formulation of system concepts to satisfy those needs identified in the SON. The SOC is a list of concepts covering such areas as hardware, software, methodologies, training and support that form the basis for a modern programming environment for DMA. The SOC is the key link between the SON and the particular implementation to be proposed for a Near-Term (1985) and Far-Term (1987) MPE.

The SOC's were formulated from the needs identified in the SON, MPE concepts, and GD/DSD experience during project personnel team meetings at DSD/Central Center in Fort Worth during the month of July, 1981. The SON/SOC matrix (see Figure 5.2) resulting from these meetings is a mapping of the operational needs identified in the SON into one or more generic programming concepts which satisfy each need.

Needs were identified as center specific when grouping showed a high need at one center and a low need at the other. All needs with a low rating at both sites inclusively have been eliminated from the SON/SOC. The columns of the matrix represent various concepts that could satisfy the particular needs. When a need is partially or completely satisfied by a concept an "X" appears at the point of intersection in the matrix. Note that a particular concept can satisfy more than one need and a particular need may require more than one concept to satisfy it.

The SON/SOC matrix has two outstanding benefits. Consistency can be traced between the SON and the SOC, and the system concepts are generic in nature which allows for more than one implementation method. The proposed alternative implementations were to be used in-part to develop the Near-Term and Far-Term Modern Programming Environment specifications for DMA.

5.0 IN-PROCESS-REVIEW OF SON/SOC

The SON/SOC matrix was presented at an In-Process-Review (IPR) conducted at RADC on 18-19 August, 1981. As a result of this review several changes were made to the SON/SOC matrix and the SON. The changes involved elimination of certain concepts (20-automated tape management procedures, 23-software scheduling package and 26-information interchange) and needs (43-better tape procedures, 45-replace/terminate DCT 2000's and 51-remote access by federal agencies) as being outside the realm of the DMA MPE study since they were not part of software development. In addition SOC11 was revised to apply more closely to current DMA needs and SOC18 was expanded to include graphics. It was also requested that the needs be categorized in some manner to improve on the readability of the matrix. This was accomplished by grouping the needs relative to the software life cycle; with some needs appearing in multiple groups. The revised SON and SON/SOC appear in Figures 5.1 and 5.2 respectively.

At a follow-up status meeting in St. Louis, certain center specific needs were eliminated because of an improved understanding of the purpose of a "DMA need". It was decided that the needs should reflect MPE needs, not necessarily center needs. Hence, certain needs that were center specific were eliminated from the SON.

09 OCTOBER 1981

```
SSSSSSSS  00000000  NN      NN
SSSSSSSS  00000000  NNN     NN
SS     SS 00      00 NNN     NN
SS        00      00 NNNN    NN
SS        00      00 NNNN    NN
SSSSSSSS  00      00 NN NN NN
SSSSSSSS  00      00 NN NN NN
       SS 00      00 NN   NNNN
       SS 00      00 NN   NNNN
SS     SS 00      00 NN    NNN
SSSSSSSS  00000000  NN     NNN
SSSSSSSS  00000000  NN      NN
```

CATEGORY	BASIC RATING DMAAC	DSD	DMAHTC
PROJECT MANAGEMENT			
10 IMPROVED MILESTONE IDENTIFICATION	3	3	5
12 IMPROVE MANLOADING	4	4	4
14 IMPROVE SCHEDULE IMPACT ANALYSIS	3	4	5
48 CHARGEBACK SYSTEM	3	5	3
56 MANAGEMENT TRACKING FUNCTIONS	3	5	3
REQUIREMENTS			
1 FORMAL REQUIREMENTS SPECIFICATION	4	3	5
5 REQUIREMENTS TRACKING	3	3	4
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
DESIGN			
21 SIMULATOR FOR DESIGN	3	3	4
22 PROGRAM DESIGN LANGUAGE	4	4	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
CODING			
55 MODERN SOURCE DATA ENTRY TECHNIQUES	5	5	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
TEST			
2 QA PROCEDURES AND GUIDELINES	3	5	5
21 SIMULATOR FOR DESIGN	3	3	4
36 GRAPHICS AIDS	5	4	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
MAINTENANCE			
9 CONFIGURATION CONTROL	5	5	5
40 HISTORICAL DATA BASE TECHNIQUES	3	3	5
57 SOFTWARE DEVELOPMENT TOOLS	5	5	5
58 PRODUCTION PROGRAM OPTIMIZATION	3	4	3
59 STANDARDIZED PHASED DEVELOPMENT	3	5	3
OTHER			
3 INTERACTIVE SYSTEM ACCESS	5	5	5
4 INCREASED NUMBER OF TERMINALS	5	5	5
11 DECREASED PAPERWORK	3	5	5
16 UPDATE OF OLD DOCUMENTATION	5	3	5
18 FASTER INTEGRATION OF NEW EMPLOYEES	3	3	5
34 AUTOMATED TEXT MANAGEMENT TOOL	3	3	5
41 ORGANIZATION TOOLS/TECHNIQUES INTERFACE	3	4	5
42 USER ASSISTANCE FUNCTION	3	4	3
44 ERROR RATE STANDARDS	3	3	3
46 REDUCE ACCOUNTING DATA REPORT ANOMALIES	3	3	3
47 COMPREHENSIVE TRAINING PROGRAM	5	5	5
52 DECREASE TURNAROUND TIME TO MINUTES	3	3	3
54 NATURAL LANGUAGE USER/SYSTEM INTERFACE	3	3	3
60 STANDARDIZED DEVELOPMENT HARDWARE	4	5	4

Figure 5.1   Current SON

NEEDS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
PROJECT MANAGEMENT																							
10 IMPROVED MILESTONE IDENTIFICATION							X		X														
12 IMPROVE MANLOADING							X	X	X			X											
14 IMPROVE SCHEDULE IMPACT ANALYSIS							X	X	X														
48 CHARGEBACK SYSTEM																					X		
56 MANAGEMENT TRACKING FUNCTIONS	X					X	X	X	X												X		
REQUIREMENTS																							
1 FORMAL REQUIREMENTS SPECIFICATION	X												X										
5 REQUIREMENTS TRACKING													X										
57 SOFTWARE DEVELOPMENT TOOLS	X	X									X		X					X		X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X			X							X			
DESIGN																							
21 SIMULATOR FOR DESIGN											X							X					
22 PROGRAM DESIGN LANGUAGE											X			X									
57 SOFTWARE DEVELOPMENT TOOLS	X	X									X			X				X		X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X				X						X			
CODING																							
55 MODERN SOURCE DATA ENTRY TECHNIQUES			X	X											X								
57 SOFTWARE DEVELOPMENT TOOLS	X	X													X			X		X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X					X					X		X	
TEST																							
2 QA PROCEDURES AND GUIDELINES										X										X			
21 SIMULATOR FOR DESIGN											X							X					
36 GRAPHICS AIDS	X			X														X					
57 SOFTWARE DEVELOPMENT TOOLS	X	X									X							X	X	X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X									X	X			
MAINTENANCE																							
9 CONFIGURATION CONTROL	X		X		X																		
40 HISTORICAL DATA BASE TECHNIQUES							X										X						
57 SOFTWARE DEVELOPMENT TOOLS	X	X			X													X	X	X			
58 PRODUCTION PROGRAM OPTIMIZATION	X																		X				
59 STANDARDIZED PHASED DEVELOPMENT	X				X					X										X			
OTHER																							
3 INTERACTIVE SYSTEM ACCESS				X																			
4 INCREASED NUMBER OF TERMINALS				X																			
11 DECREASED PAPERWORK	X		X			X	X									X	X						
16 UPDATE OF OLD DOCUMENTATION	X																						
18 FASTER INTEGRATION OF NEW EMPLOYEES		X								X		X			X								X
34 AUTOMATED TEXT MANAGEMENT TOOL	X					X									X	X							
41 ORGANIZATION TOOLS/TECHNIQUES INTERFACE		X	X	X	X								X	X									
42 USER ASSISTANCE FUNCTION					X																		X
44 ERROR RATE STANDARDS																					X		X
46 REDUCE ACCOUNTING DATA REPORT ANOMALIES																	X				X		
47 COMPREHENSIVE TRAINING PROGRAM										X		X											
52 DECREASE TURNAROUND TIME TO MINUTES				X																	X		
54 NATURAL LANGUAGE USER/SYSTEM INTERFACE														X									
60 STANDARDIZED DEVELOPMENT HARDWARE			X	X																			

NEEDS

SON / SOC MATRIX

CONCEPTS

1. INTEGRATED SUPPORT DEVELOPMENT SYS
2. HIGH ORDER LANGUAGE
3. SINGLE LARGE MULTI-USER ENVIRONMENT
4. STANDARD SMALL MULTIPLE ENV'S
5. CONFIGURATION CONTROL SYSTEM
6. AUTOMATED OFFICE
7. PROJECT MANAGEMENT SYSTEM
8. COST ESTIMATING SYSTEM
9. PROJECT PATH ANALYSIS METHOD
10. SOFTWARE ENGINEERING PRACTICES TRNG
11. RAPID PROTOTYPING
12. AUTOMATED TRAINING PROGRAM
13. AUTOMATED REQUIREMENTS GENERATION
14. SOFTWARE DESIGN LANGUAGE
15. STRUCTURED PROGRAMMING FACILITY
16. INTERACTIVE TEXT PROCESSING
17. AUTOMATED DATA COLLECTION
18. INTERACTIVE SUPPORT SIMULATION SOFT
19. SOFTWARE TESTING SYSTEM
20. SOFTWARE STANDARDIZATION
21. CHARGEBACK SYSTEM
22. STRUCTURED PROGRAMMING
23. USER ASSISTANCE FUNCTION

GD/DSD
Central Center

Figure 5.2 SON/SOC Matrix

6.0 NEEDS

The following paragraphs define in detail the needs as listed in the revised SON (Figure 5.1) and their applicability to DMA. First a definition of the need, which may include amplification with respect to the DMA environment, is given followed by a list of documents in which the needs were identified. For further identification of related documents, the number in parenthesis refers back to the list in Section 2.0. Page and paragraph numbers are included. Implementation priority is then provided.

6.1 DEFINITION AND ORIGIN OF NEEDS

The following items were found to be medium to high needs at both DMAHTC and DMAAC. First the need is defined, then the origin is given. The need was identified through supporting DMA documentation, through interviews with DMA personnel or through a survey questionnaire distributed to management and technical personnel at both centers.

6.1.1 (SON#1) Formal Requirements Specification: a means of formally documenting the elemental requirements of a task or project prior to the beginning of design. The method used may be a manual or automated method involving the use of a software requirements tool. Examples of these tools are USE.IT, SADT, PSL/PSA, LARE and FAME.

> ORIGIN:PAGES
> DMA MPE STUDY(7): 30,36
> GD SURVEY/QUESTIONS(11): I.1.B, I.1.R, I.2.H, I.2.I, I.2.J, I.2.R, I.2.S, III.G
> Personnel Interviews by GD/DSD(12)

6.1.2 (SON#2) Quality Assurance Procedures and Guidelines: ways of enforcing a required set of programming practices/standards covering all phases of the programming lifecycle. This provides for both better quality and consistency in software development and, therefore, more easily maintained software.

> ORIGIN:PAGES
> FEDSIM REVIEW - DMAHTC(1): 19,32
> PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-4
> DMA MPE STUDY(7): 36-40
> GD SURVEY/QUESTIONS(11): I.2.P, I.2.R, II.2.C
> Personnel Interviews by GD/DSD(12)

6.1.3 (SON#3) Interactive System Access: the ability to access the computer system through an on-line environment as opposed to card readers or over-the-counter entry stations. Output may also be accessed without necessarily being printed out on paper. This results in more freedom of access, faster turnaround time and a decrease in the amount of paper produced.

ORIGIN:PAGES
DMA OPERATIONAL CONCEPTS(2): 3, 31
DMA MPE STUDY(7): 47
GD SURVEY/QUESTIONS(11): I.1.A, I.1.C, I.1.F, I.1.G, I.1.T, I.2.L,IV.A
Personnel Interviews by GD/DSD(12)

6.1.4 (SON#4) An Increased Number of Terminals: a requirement at DMA centers if interactive access is to be made available to all programmers. Currently there are a minimal number of terminals available through which the programmers may obtain this access.

ORIGIN:PAGES
Personnel Interviews by GD/DSD(12)

6.1.5 (SON#5) Requirements Tracking: a means of documenting the coverage of and changes to the requirements of a program or system through its complete lifecycle. As with requirements specification this may be done through a standardized manual method or a commonly used automated method (software tool).

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): I.1.B, I.2.F, I.2.H, I.2.R, I.2.S

6.1.6 (SON#9) Configuration Control: the ability to track and maintain a history of changes to a system or program. Within DMA programs are commonly sent between organizations or centers. As changes occur to these programs they are not necessarily made to all production versions and eventually the program may no longer be a common system to all users. A configuration control system would keep track of these versions by the use of version or release numbers and maintain a history of the changes required to get from one version to another; thus improving communications between users of a system and providing consistency in the use of a system.

ORIGIN:PAGES
PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-8
DMA MPE STUDY(7): 48-58
GD SURVEY/QUESTIONS(11): I.1.C, I.1.L, I.2.B, II.1.C, II.1.H, IV.A

6.1.7 (SON#10) Improved Milestone Identification: a means of improving the identification and documentation of significant events in the development of a system or program. This provides an overall, high level view of a system's development process.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): I.2.A, I.2.G, I.2.M, II.1.D
Personnel Interviews by GD/DSD(12)

6.1.8 (SON#11) Decreased Paperwork: a need to lower the amount of paperwork produced at each center including computer runs and manually produced documentation associated with software development.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): I.1.J, I.2.M, II.1.G, II.2.A, II.2.E, III.H
Personnel Interviews by GD/DSD(12)

6.1.9 (SON#12) Improve Manloading: improvement of the methods of determining the amount of manpower required for a given project through manual or automated methods using parametric or historical data.

ORIGIN:PAGES
Personnel Interviews by GD/DSD(12)
GD SURVEY/QUESTIONS(11): I.1.H, I.2.E, II.1.D, IV.A

6.1.10 (SON#14) Improve Schedule Impact Analysis: improvement of the methods of determining how changes to a project will affect its schedule through automated or manual methods usually associated with identifying critical and affected paths in the development process.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): I.2.C, I.2.D, II.1.D, IV.A
Personnel Interviews by GD/DSD(12)

6.1.11 (SON#16) Update of Old Documentation: improvement of the documentation associated with existing programs available for maintenance purposes.

ORIGIN:PAGES

Personnel Interviews by GD/DSD(12)

6.1.12 (SON#18) Faster Integration of New Employees: to provide assistance in the training and orientation of new employees into the programming environment of DMA and its associated standards and methodology for software development.

ORIGIN:PAGES
Personnel Interviews by GD/DSD(12)

6.1.13 (SON#21) Simulator for Design: a system of software tools which would enable the rapid prototyping of a production environment in order to verify the basic design of developed software.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): IV.A

6.1.14 (SON#22) Program Design Language: a language used in the design and documentation of complex software applications.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): I.2.K, I.2.S, IV.A

6.1.15 (SON#34) Automated Text Management System: a software system which would provide basic support in the development of textual material associated with the software development process including such functions as sorting, merging, copying, formatting and archiving; as well as the capabilities associated with text editing tools.

ORIGIN:PAGES
DMA OPERATIONAL CONCEPTS(2): 8
GD SURVEY/QUESTIONS(11): IV.A
Personnel Interviews by GD/DSD(12)

6.1.16 (SON#36) Graphics Aids: hardware and/or software which would provide the capability to display plotter type information in an interactive CRT format or change data from one format to another for display.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): IV.A
Personnel Interviews by GD/DSD(12)

6.1.17 (SON#40) Historical Data Base Techniques: methods, either manual or automated, of collecting information and

statistics associated with software development activities to provide a basis for evaluation of future tasks.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): I.2.C, IV.A

6.1.18 (SON#41) Organizational Tools/Techniques Interface: a means of providing a common format for the exchange of ideas and information between organizations within DMA which have a functional dependency.

ORIGIN:PAGES
GD SURVEY/QUESTIONS(11): IV.A
Personnel Interviews by GD/DSD(12)

6.1.19 (SON#42) User Assistance Function: a method to assist users in overcoming and avoiding errors. The user assistance function people would not be expected to help users debug their programs but would help all users who had production run problems. Additional duties would be conducting error rate studies, conducting meetings with users explaining how to avoid errors, disseminating information on the better use of the computer system, including a system change bulletin, and augmenting the information flow to management so they may respond more quickly to user needs.

ORIGIN:PAGES
FEDSIM REVIEW - DMAHTC(1): 24-25
FEDSIM REVIEW - DMAAC(6): 37, 49-50
FEDSIM ERROR RATE STUDIES(8): 42

6.1.20 (SON#44) Error Rate Standards: the formulation of limits on specific error repetitions, possibly within a given time frame. Reports would be generated on these errors and sent to all organizations. Corrective action can then be taken by each organization for areas where limits are exceeded or justification provided for exceeding the limit. A method of revising the limits must be included in the standards.

ORIGIN:PAGES
DMA MPE STUDY(7): 43-45, 47
FEDSIM ERROR RATE STUDIES(8): 42-43

6.1.21 (SON#46) Reduced Accounting Data Report Anomalies: DMA's accounting data is a conservative indicator of the overall error rate. The accounting file reports more erroneous runs as good than good runs as erroneous. These erroneous reports should be reduced in number.

ORIGIN:PAGES
FEDSIM ERROR RATE STUDIES(8): 41-42

6.1.22 (SON#47) Comprehensive Training Program: training to provide personnel with a background in software development techniques, requirements specification, design, testing, standard practices, project planning, estimating and scheduling.

ORIGIN:PAGES
FEDSIM REVIEW - DMAHTC(1): 24, 26-27, 33
DMA OPERATIONAL CONCEPTS(2): 25
PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-14, 2-15
FEDSIM REVIEW - DMAAC(6): 29-31, 59
DMA MPE STUDY(7): 58-71, 82-95
GD SURVEY/QUESTIONS(11): II.2.A, III.F

6.1.23 (SON#48) A Chargeback System: a system which assigns charges to each unit of computer usage by user and by run such that each user run has a unit charge associated with it. The real aim of the system is not so much to allocate costs as to create the proper incentives for the users to become involved in ADP management and to conserve their use (i.e., use fewer tapes, run fewer jobs, and make those jobs more efficient) to enable the computer facility to provide responsive, efficient service.

ORIGIN:PAGES
FEDSIM REVIEW - DMAHTC(1): 14, 31
FEDSIM REVIEW - DMAAC(6): 24-25, 54

6.1.24 (SON#52) To Decrease Turnaround Time to Minutes: both centers are experiencing lengthy turnaround times on the mainframe computers. A decrease in turnaround time to minutes for the predominate number of runs will be required. Currently next day batch service is normal.

ORIGIN:PAGES
DMA OPERATIONAL CONCEPTS(2): 58
PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-4
FEDSIM REVIEW - DMAAC(6): 37
DMA MPE STUDY(7): 43, 47
GD SURVEY/QUESTIONS(11): I.1.I, II.2.A

6.1.25 (SON#54) Natural Language User/System Interface: a system which would interface the user to the computer in a manner which is less constrained in syntax and semantics than normal control and algorithmic languages.

ORIGIN:PAGES

DMA OPERATIONAL CONCEPTS(2): 13, 31

6.1.26 (SON#55) Modern Source Data Entry Techniques: the capability to enter data into a computer through the most efficient means available matching the form of the data to be entered, for example, disk, floppy, source, binary, cards, tape and the systems available, i.e., CRT, RJE, card reader, disk/tape drive, etc.

ORIGIN:PAGES
DMA OPERATIONAL CONCEPTS(2): 34

6.1.27 (SON#56) Management Tracking Functions: processes available to project managers which provide cost analysis, budget tracking, schedule impact information and report generation capabilities.

ORIGIN:PAGES
FEDSIM REVIEW - DMAHTC(1): 12
DMA OPERATIONAL CONCEPTS(2): 32
PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-8
DMA MPE STUDY(7): 48

6.1.28 (SON#57) Software Development Tools: computer programs which enable the user to perform activities in the life cycle development of software without extensive training and/or which decrease the amount of manual labor associated with the activity; an example being one high order language for several computers. High order languages are easier to learn than assembly languages and may be common to several architectures. Most of the tools will guide a person through the steps of a process; hence extensive training in an area such as requirements specification would not be required. At the same time most tools provide automatic documentation and analysis of its task.

ORIGIN:PAGES
DMA OPERATIONAL CONCEPTS(2): 31
PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-10
DMA MPE STUDY(7): 71-78
GD SURVEY/QUESTIONS(11): I.2.L, IV.A

6.1.29 (SON#58) Production Program Optimization: optimization of programs which have high computer resource requirements through the use of automated tools to identify structures or code which could be modified to decrease the programs effects on the production environment resources.

ORIGIN:PAGES
FEDSIM REVIEW - DMAHTC(1): 18, 33

PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-5
DMA MPE STUDY(7): 43-47
FEDSIM ERROR RATE STUDIES(8): 41

6.1.30 (SON#59) Standardized Phased Development: development of software in a life cycle phased methodology consistently across the DMA organization. This would include standardization of programming tools and techniques, documentation and configuration control.

ORIGIN:PAGES
Personnel Interviews by GD/DSD(12)

6.1.31 (SON#60) Standardized Development Hardware: common development hardware used throughout DMA to maximize the portability of development tools, increase the efficiency of any configuration control system and decrease the training required by the use of diversified architectures.

ORIGIN:PAGES
PROGRAM SUPPORT LIBRARY - INTERIM REPORT(3): 2-8, 2-9, 2-10
FEDSIM REVIEW - DMAAC(6): 47, 60

6.2 PRIORITY OF NEEDS

This grouping of needs is a priority list of the needs which are expressed in the SON. The first group has the highest priority, the last group the least. The data used to develop this list includes information gathered during the tool evaluation phase, October - November, 1981; general knowledge of the DMA environment; and the need for a smooth transition during implementation of solutions. Rationale is included for the grouping of needs generated by GD/DSD, DMAHTC and DMAAC. The needs given the highest priority will be those addressed first when transitioning from the current DMA environment to the near-term and subsequent far-term environments. Implementing, as possible, the solutions to the highest ranked priorities first will result in more immediately apparent benefits during the implementation of the MPE. Additionally, use of the rankings will assure the most thorough coverage within the recommended MPE of the most critical needs within DMA.

6.2.1 Comparison by Group

First groups are identified, followed by the needs which an organization perceived to fall into the categories. The data following paraphrases the rationale by which the organization

prioritized the needs. The last set of grouped needs is a numerical average of the five rankings submitted.

GD/DSD GROUPING

GROUP 5: 3,4
GROUP 4: 52,55,58,60
GROUP 3: 1,2,9,22,34,41,48,57,59
GROUP 2: 5,10,12,14,18,42,44,46,47,56
GROUP 1: 11,16,21,36,40,54,

Group 5 -- provide users with an interactive access capability to current development hardware.
Group 4 -- provide the users with a near-term hardware/software support system consisting of immediately available tools.
Group 3 -- integrate and modified as necessary to provide a consistant and environmentally compatible methodology for software development.
Group 2 -- a partially parallel effort to provide training and management support must be implemented.
Group 1 -- advanced support systems should be provided and old systems upgraded or replaced.

DMAAC GROUPING

GROUP 5: 3,4
GROUP 4: 22,34,36,41,42,47,52,55,57,58,60
GROUP 3: 1,2,9,10,11,12,14,40,44,46,48,56,59
GROUP 2: 5,16,18,21
GROUP 1: 54

Group 5 -- increase programmer productivity by providing quicker access and improved response time.
Group 4 -- provide hardware/software system support, tool integration, and standards.
Group 3 -- provide advanced automated management support tools.
Group 2 -- provide support for old programs and new developments.
Group 1 -- analyze advanced techniques/capabilities.

DMAHTC GROUPING

GROUP 5: 3,4,52,55
GROUP 4: 2,9,16,22,34,36,41,42,57,58,59
GROUP 3: 10,12,14,18,44,46,47,48,56
GROUP 2: 1,5,21,40
GROUP 1: 11,54,60

Group 5 -- the need to improve programmer productivity and user access.

Group 4 -- the need to improve management of software projects through establishment of standards, tools and procedures; software/hardware support.

Group 3 -- the need to improve general administrative management through better resource allocation, scheduling and accounting.

Group 2 -- the need to provide the capability for system definition and design.

Group 1 -- the need for advanced software/technical support.

DMAHQ GROUPING

GROUP 5: 2,3,4,9,40,47,48,55,56,57,59,60
GROUP 4: 12,14,16,36,41,42,44,58
GROUP 3: 1,5,10,11,18,22,34,46,52
GROUP 2: 21
GROUP 1: 54

RADC GROUPING

GROUP 5: 3,9,34,47,55,57,60
GROUP 4: 10,22,36,42,58
GROUP 3: 2,4,16,46,48,54,56,59
GROUP 2: 1,5,11,12,40,41
GROUP 1: 14,18,21,44,52

AVERAGE GROUPING

GROUP 5: (4.3-5.0) 3,4,55
GROUP 4: (3.5-4.2) 2,9,22,34,42,47,57,58,59,60
GROUP 3: (2.7-3.4) 1,10,12,16,36,41,46,48,52,56
GROUP 2: (1.9-2.6) 5,11,14,18,40,44
GROUP 1: (1.0-1.8) 21,54

6.2.2 Comparison by Need

GD/DSD, RADC, DMAHQ, DMAAC and DMAHTC personnel supplied inputs to help define a priority weighting factor for each need from the SON. These are on a scale of 5-1 with a 5 indicating the greatest need and 1 the least. These weighting factors were used in computing the total score for an implementation of a concept as described in Section 13.0.

ABBREVIATIONS: DSD-G,DMAAC-A,DMAHTC-H,RADC-R,DMAHQ-Q

		G	A	H	R	Q	avg.
1	FORMAL REQUIREMENTS SPECIFICATION	3	3	2	2	3	2.6
2	QA PROCEDURES AND GUIDELINES	3	3	4	3	5	3.6
3	INTERACTIVE SYSTEM ACCESS	5	5	5	5	5	5.0
4	INCREASED NUMBER OF TERMINALS	5	5	5	3	5	4.6

5	REQUIREMENTS TRACKING	2	2	2	2	3	2.2
9	CONFIGURATION CONTROL	3	3	4	5	5	4.0
10	IMPROVE MILESTONE IDENTIFICATION	2	3	3	4	3	3.0
11	DECREASED PAPERWORK	1	3	1	2	3	2.0
12	IMPROVE MANLOADING	2	3	3	2	4	2.8
14	IMPROVE SCHEDULE IMPACT ANALYSIS	2	3	3	1	4	2.6
16	UPDATE OF OLD DOCUMENTATION	1	2	4	3	4	2.8
18	FASTER INTEGRATION OF NEW EMPLOYEES	2	2	3	1	3	2.2
21	SIMULATOR FOR DESIGN	1	2	2	1	2	1.6
22	PROGRAM DESIGN LANGUAGE	3	4	4	4	3	3.6
34	AUTOMATED TEXT MANAGEMENT TOOL	3	4	4	5	3	3.8
36	GRAPHICS AIDS	1	4	4	4	4	3.4
40	HISTORICAL DATA BASE TECHNIQUES	1	3	2	2	5	2.6
41	ORGANIZATIONAL TOOLS/TECHNIQUES INTERFACE	3	4	4	2	4	3.4
42	USER ASSISTANCE FUNCTION	2	4	4	4	4	3.6
44	ERROR RATE STANDARDS	2	3	3	1	4	2.6
46	REDUCE ACCOUNTING DATA REPORT ANOMALIES	2	3	3	3	3	2.8
47	COMPREHENSIVE TRAINING PROGRAM	2	4	3	5	5	3.8
48	CHARGEBACK SYSTEM	3	3	3	3	5	3.4
52	DECREASE TURNAROUND TIME TO MINUTES	4	4	5	1	3	3.4
54	NATURAL LANGUAGE USER/SYSTEM INTERFACE	1	1	1	3	1	1.4
55	MODERN SOURCE DATA ENTRY TECHNIQUES	4	4	5	5	5	4.6
56	MANAGEMENT TRACKING FUNCTIONS	2	3	3	3	5	3.2
57	SOFTWARE DEVELOPMENT TOOLS	3	4	4	5	5	4.2
58	PRODUCTION PROGRAM OPTIMIZATION	4	4	4	4	4	4.0
59	STANDARDIZED PHASED DEVELOPMENT	3	3	4	3	5	3.6
60	STANDARIZED DEVELOPMENT HARDWARE	4	4	1	5	5	3.8

7.0 DEFINITION OF CONCEPTS

In the following subparagraphs each of the generic concepts identified in the SON/SOC matrix is defined and rationale is provided for satisfying the indicated needs.

7.1 SOC 1 INTEGRATED SUPPORT DEVELOPMENT SYSTEM

DEFINITION: An integrated set of tools developed to support all or part of the software development life cycle, i.e., requirements, design, coding, testing, and maintenance; and to support management functions, usually written in the language supported.

SON ITEMS SUPPORTED:1,9,11,16,34,36,56,57,58,59

An integrated set of tools supporting the entire life cycle of software development would include a methodology which could be formalized as a DMA standard. A configuration control subsystem would be an integral part of the life cycle toolset. Other tools in the system would include an automated text management system to support programming and graphics aids to support testing. These other tools would also support project management functions. Paperwork would be decreased due to life cycle phases being supported interactively as opposed to manually; old documentation could be updated by processing old programs through tools which provide documentation as part of their outputs. These outputs would also be very useful in optimizing production programs by identifying current capabilities and complexities associated with their execution. All of the tools and techniques for utilization could be adopted or molded to conform to a standardized phased development system for DMA.

7.2 SOC 2 HIGH ORDER LANGUAGE

DEFINITION: A language in which each instruction or statement corresponds to several machine code instructions; allowing users to write in a notation with which they are familiar, independent of hardware. The primary examples under consideration are FORTRAN, COBOL, and Ada.

SON ITEMS SUPPORTED:18,41,57

A high order language is a software development tool supporting the programming phase. High order languages are used because they express a procedure and the data being manipulated in a format closer to common language and mathematics than would have to be used if assembly or machine

code were utilized. This format allows a person to more quickly learn how to use a computer because the hardware is not addressed in the languages. The use of a high order language hosted on multiple architectures provides a communications interface for expressing problems between different organizations working with different machines or applications.

7.3 SOC 3 SINGLE LARGE MULTI-USER ENVIRONMENTS

DEFINITION: Uniform single system tool bearing hosts with remote job entry (RJE) stations and the capability to support multiple varieties and a large number of interactive terminals.

SON ITEMS SUPPORTED:9,11,41,55,60

A single large standardized software development environment at each center would help simplify the configuration control problem that exists. Total automation of the system could be achieved and multiple copies and configurations of software would be easier to track and manage. Most large systems usually support advanced word processing capabilities as well as mail functions and report generation facilities. These capabilities decrease the amount of paperwork generated manually in intermediate and final form. A large system would also allow all departments to use the same programming support environment which would provide a common interface to libraries, tools, information distribution, etc.

7.4 SOC 4 STANDARD SMALL MULTIPLE ENVIRONMENTS

DEFINITION: Uniform small, identical computer systems on which to perform software development each supporting multiple interactive terminals.

SON ITEMS SUPPORTED:3,4,36,41,52,55,60

A standard configuration of small software development environments would increase the physical interactive access capabilities by distributing the support terminals by functional responsibility over a wider area. This could be accomplished with less effort than distributing terminals from a central site. Response time generally is decreased with the use of small systems, especially when a large number of terminals are to be supported. An additional benefit of this type configuration is that when one system is down for maintenance or a scheduled priority job, other systems can pick up the work load. Most minicomputer systems support graphics packages which would allow the user to generate

program output in his work area for analysis before putting the software into production. The standardization of the configurations would provide a common interface for communication of programs and ideas between systems, functional areas and/or centers, as well as provide guidance to future procurements with respect to the hardware/software interfaces required. These systems could interface to the production mainframes as front ends allowing source data to be entered through CRT, tape, disk, or cards, as appropriate.

7.5 SOC 5 CONFIGURATION CONTROL SYSTEM

DEFINITION: An automated system to track and maintain a history of changes to a system or program through development and maintenance life cycles.

SON ITEMS SUPPORTED:9,41,42,57,59

By definition this type of system would provide a means of maintaining configuration control over the software releases produced. Such a system would also provide information to a user assistance function accurately and automatically which could then be distributed to users in all organizations on the latest updates in software. This type of software tool could be used as part of a standardized phased development system.

7.6 SOC 6 AUTOMATED OFFICE

DEFINITION: Using computers to perform as many typical office tasks as possible.

SON ITEMS SUPPORTED:11,34,56

An automated office system usually includes interactive capabilities to send and receive messages, to generate correspondence and documentation using word processors and to invoke basic mathematical functions. Such systems also include hardware to support multiple output formats. This type of system would decrease the amount of paperwork generated, provide for an automated means of text management, and supply management with report generation capabilities.

7.7 SOC 7 PROJECT MANAGEMENT SYSTEM

DEFINITION: Automated assistance in effective project planning, scheduling, monitoring and control.

SON ITEMS SUPPORTED:10,11,12,14,40,56

Milestone identification, manloading projections and schedule impact analysis are all part of project management systems. They allow a manager to control and analyze his projects while generating a history of the activities as updates to project plans are encountered. The paperwork associated with the functions is reduced through interactive access and magnetic storage of intermediate data.

7.8 SOC 8 COST ESTIMATING SYSTEM

DEFINITION: A system used to evaluate software costs associated with a given project by assessing the behavior of the variables which impact life cycle cost and investigating the project's sensitivities to parameter changes.

SON ITEMS SUPPORTED:12,14,56

Two of the inputs into a cost estimating system for software development are manloading and schedule. A model is built which can be analyzed by modifying the input variables to bracket the original values. Using these techniques a manager can improve his evaluations of schedule impacts and manloading requirements. As a project evolves, the manager can verify/modify his inputs to provide historical data for future projections and to update current projections. This system would be applied across all organizations and specific "factors" developed for each which would give an indication of the complexity, size, volume, etc., of the software developed.

7.9 SOC 9 PROJECT PATH ANALYSIS METHOD

DEFINITION: A method to organize project components, monitor their progress and display their status graphically.

SON ITEMS SUPPORTED:10,12,14,56

Milestone identification, schedule, and manloading requirements are project components which must be analyzed and monitored during the software development life cycle. Using project path analysis methods such as CPM or PERT, these components can be graphically displayed and tracked, providing a high level visual source of data for management.

7.10 SOC 10 SOFTWARE ENGINEERING PRACTICES TRAINING

DEFINITION: A training program to establish standard practices for software development and effective utilization of human and computer resources.

SON ITEMS SUPPORTED:2,18,47,59

One task in software engineering is to identify a standardized phased development process to be utilized in a specific environment. Once the process is established, training is required to provide the users with the specifics of the process which vary from known textbook methods or from another environment with which personnel may be familiar. This training is part of any comprehensive training program designed to provide quality assurance to products being produced or to help integrate new employees into a programming environment.

7.11 SOC 11 RAPID PROTOTYPING

DEFINITION: A methodology used to define programs or describe program attributes in a high level, possibly in non-procedural form, to provide the capability of modeling an environment for analysis or constructing a non-production program from component parts.

SON ITEMS SUPPORTED:21,22,57

The design phase of the software life cycle is the current area of interest in many academic communities; but the main area of study is methodologies, which provide no interactive support for current systems. To support users interactively, these methodologies need to be supported through software development tools which verify their use. Additionally these tools should be able to support simulation of the design using some form of program design language.

7.12 SOC 12 AUTOMATED TRAINING PROGRAM

DEFINITION: An interactive, self-paced, computer assisted program fully contained and user friendly to be used by an individual familiar enough with his environment to be able to identify entry points into the system.

SON ITEMS SUPPORTED:12,18,47

An automated training program which is self-paced allows employees to learn at a rate which is optimum for their experience and background. Such programs also have a minimal effect on an employee's job responsibilities by allowing him to train as time and work permit. This training should not include a topic that <u>all</u> employees will need information about since this would put a continuous load on one source. This type of general training would be covered by the program defined in SOC 10. Automated training should be used for

advanced or specialized topics and should be a part of any comprehensive training program.

7.13 SOC 13 AUTOMATED REQUIREMENTS GENERATION

DEFINITION: A software capability which can create a requirements data base relating to the specification of a system which may be analyzed for consistency, completeness and traceability.

SON ITEMS SUPPORTED:1,5,41,57,59

An automated requirements generation capability could be formalized to provide a requirements specification method for DMA or a capability could be developed to conform to current DMA practices. This capability, or tool, could then be used to specify and track requirements and changes to requirements as a project progresses. A formalized method would serve as a communications interface between organizations requiring and providing support and could be standardized as part of a phased development scenario.

7.14 SOC 14 SOFTWARE DESIGN LANGUAGE

DEFINITION: A software design methodology and associated system which provides an effective communications medium to support the design and documentation of complex software applications.

SON ITEMS SUPPORTED:22,41,54,57,59

A software design language should include a programming design language and supporting software to verify use and provide documentation. These systems are utilized to allow communication at a high level (more English-like) between manager and designer, at a low level (more HOL-like) between designer and programmer, and additionally between designers. This system should be a part of the standardized phased development of software supporting the design life cycle phase.

7.15 SOC 15 STRUCTURED PROGRAMMING FACILITY

DEFINITION: A menu driven collection of software development tools which includes a text editor and library maintenance utilities designed to reduce keystrokes and the opportunity for error.

SON ITEMS SUPPORTED:18,34,55,57,59

A structured programming facility is an interactive software development tool which supports modern source data entry techniques, automated text management, and system utilities to perform common functions such as sorting, searching, and basic math functions. The system is usually menu driven providing for a short learning period to effectively exercise basic applications. This type of system could be implemented in support of the coding task of a standardized phased development system.

7.16 SOC 16 INTERACTIVE TEXT PROCESSING

DEFINITION: An automated, interactive system to build, print, edit, store and retrieve textual data, possibly including the ability to perform basic tabulating and arithmetic functions.

SON ITEMS SUPPORTED:11,34

Interactive text processing, or word processing, can be used to decrease paperwork where typing support is required through the use of one-step function keys which modify blocks of text, by saving previous copies of text which can be easily modified and by magnetically archiving documentation. This task is one part of an automated text management system.

7.17 SOC 17 AUTOMATED DATA COLLECTION

DEFINITION: The ability to interactively assimilate and maintain information in a chronologically dependent format.

SON ITEMS SUPPORTED:11,40,46

An automated means of collecting data could be used to decrease the amount of paper generated by collecting the data in magnetic storage. The system could also be utilized in a mode that would keep historical records rather than overwriting previous data.

7.18 SOC 18 INTERACTIVE SUPPORT SIMULATION SOFTWARE

DEFINITION: A related set of software tools which simulate the environment under which an operational program will execute by representing certain features of a physical or abstract system.

SON ITEMS SUPPORTED:21,36,57

Simulation software should be a part of the software development tool set. This software should be able to simulate the production environment subset with which the

software under development would interface, providing a means of testing the software in the programming environment. The substantial graphics environment of DMA should be a prime consideration in defining the simulation interfaces and tool output.

7.19 SOC 19 SOFTWARE TESTING SYSTEM

DEFINITION: A program analysis system used to evaluate software and demonstrate fulfillment of documented needs. Included are automatic test generators, data base analyzers, dynamic analyzers, static analyzers, test managers, etc.

SON ITEMS SUPPORTED:57,58,59

A standardized phased development process must include a testing methodology as a software tool. The methodology should be automated and applied to both internally developed and customer supplied software. Metrics should be generated, either manually or by the testing system, which could be used to gauge the software delivered. Additionally the tool could be used on old software to identify areas of possible optimization, especially when an old program is tested against new data.

7.20 SOC 20 SOFTWARE STANDARDIZATION

DEFINITION: An aid for programmers in writing and checking program documentation/code and managers for quality assurance.

SON ITEMS SUPPORTED:2,57,59

Software standardization is a specification of the model into which a program should fit. Part of the specification is quality assurance standards identified by an organization. Other parts include documentation to be produced and testing procedures to be followed. This specification should be a part of any standardized phased development plan.

7.21 SOC 21 CHARGEBACK SYSTEM

DEFINITION: A means of keeping precise account of the resources used by a user to create incentives for users to conserve resources.

SON ITEMS SUPPORTED:44,46,48,52,56

This type system would provide managers with information that could be used in conjunction with error rate standards to identify needs within an organization. An additional benefit is that accounting data would be more detailed and less likely to be erroneously reported. With an increase in management visibility of computer resource allocations an incentive should be created to conserve utilization. This would help decrease turnaround time.

7.22 SOC 22 STRUCTURED PROGRAMMING

DEFINITION: A style of programming in which the structure of a program is made as clear as possible by using three control logic structures: sequence, selection and iteration.

SON ITEMS SUPPORTED:58,59

The use of structured programming practices results in programs that are readable and easily modified, hence easily maintained. In the case of existing non-structured programs the improved capability may not be worth the effort of modification. For original programs that are not real-time or time critical, structured programming should be a part of a standardized phased development scenario.

7.23 SOC 23 USER ASSISTANCE FUNCTION

DEFINITION: An organizational function to assist users in overcoming and avoiding errors, conduct error rate studies and augment the information flow to management enabling quicker response to user needs.

SON ITEMS SUPPORTED:18,42,44

A user assistance function would help integrate new employees faster by identifying, through trend analysis, problems they would be likely to encounter and inserting them into the training curriculum. For the unusual problems encountered once a person is trained, the function would help to speed their resolution. Part of the function would be to conduct error rate studies to discover trends which might be useful to management concerning training, development, and production.

8.0 TOOL SURVEY

Another part of the Interactive Computer Program Development System Study was the evaluation and selection of software tools for the DMA MPE. The tool selection consisted of two phases. Phase I entailed the gathering of data on the software tools available in the near future and their applicability to the DMA programming environment. Phase II had as its main activity an in-depth evaluation of specific tools representing the classes of tools covering the software life cycle in a simulated Defense Mapping Agency (DMA) development scenario; as well as an analysis of factors specific to DMA which could constrain the use of tools or identify R&D efforts to enhance their capabilities.

The following sections detail the activities which were conducted during both phases of the tool survey as well as the resulting conclusions and related documentation.

The tool survey was an integral part of the task to develop a Functional Description and System/Subsystem Specification for a modern programming environment for the Defense Mapping Agency.

9.0 PHASE I ACTIVITIES

The primary goal of phase I was the gathering of data about software tools. A multi-step process was followed:

1. Literature search
2. Selection based on DMA environment
3. Tool demonstrations
4. Analysis of demonstration results.

The results of this method provided the basis for the selection of tools for the in-depth evaluation phase (phase II).

9.1 TOOL INFORMATION SOURCES

The first activity involved a literature search for software tools applicable to the Defense Mapping Agency software development environment. Software tool directories and inputs from DMA and RADC personnel were the major sources of information.

The following tool information sources were used:

1) Tools Fair - 5th International Conference on Software Engineering
2) American Institute of Aeronautics and Astronautics (AIAA)/Grumman Software Tool Survey
3) National Bureau of Standards (NBS) Software Data Base
4) Digital Equipment Corporation (DEC) Referral Catalog
5) On-Line-Systems Catalog
6) General Dynamics Tools Directory for Embedded Systems
7) Sperry Univac 1100 Series Scientific Software
8) Tutorial: Automated Tools for Software Engineering
9) Tutorial: Software Design Techniques
10) Reifer Consultants, Inc. - Software Tool Directory
11) Automated Tools for Software Engineering Seminar
12) Conversion Products/Aids Survey
13) RADC
14) Defense Mapping Agency Headquarters (DMAHQ)
15) Defense Mapping Agency Aerospace Center (DMAAC)
16) Defense Mapping Agency Hydrographic/Topographic Center (DMAHTC).

9.2 DMA TOOL APPLICABILITY

Applicability of tools to the DMA programming environment was based upon several characteristics. These were selected after analysis of the software requirements of DMA through personnel interviews, a questionnaire survey (Appendix A) and input from the following government related documents:

1) FEDSIM (Federal Computer Performance Evaluation and Simulation Center) Installation Review - DMAHTC - November 1980
2) DMA Operational Concepts (1982 - 1990) - May 1979
3) DMA Program Support Library (PSL) Interim Evaluation Report, IBM/FSD - November 1980
4) DMAAC/Scientific Computer Division - Software Life Cycle Standards - February 1981
5) DMAAC Organizational Mission Functions - October 1980
6) FEDSIM Installation Review - DMAAC - August 1980
7) DMA Modern Programming Environment (MPE) - January 1980
8) FEDSIM Optimization and Error Rate Studies - Feb 1981
9) Operational Improvement Opportunities for UNIVAC 1100/80
10) DMAHTC Organizational Manual.

The characteristics are defined as follows:

1) Portability:

The capability of a tool to be easily rehosted to a new architecture. DMA uses many different computer systems in their production environment.

2) Public domain:

Any tool developed under government funding. These tools would be available to DMA at minimal costs.

3) FORTRAN compatible*:

FORTRAN is the primary language used by DMA in scientific computing. Tools chosen should be able to analyze FORTRAN code.

*COBOL is also to be considered, but not during the tool evaluation phase of the project, as recommended during the In-Process-Review (IPR) held at RADC on 18-19 August 1981. The tools which are available or are under development and designed to work with COBOL will be analyzed by GD/DSD for the near-term and far-term environments.

4) UNIVAC hosted:

UNIVAC is the primary production hardware used and was, therefore, originally planned to be used extensively in the near-term MPE. Any tool available on this system would not require rehost effort.

5) Maturity:

Maturity implies that the tool has been in use for some time. DMA would not have many problems with this type tool in the area of debugging development errors in its code.

6) User friendly:

This characteristic provides for a short learning curve in tool usage. DMA would be able to determine the effects of tool usage on their environment without a long delay caused by training requirements.

7) DEC hosted:

Hosted on DEC hardware. Many of the varied computer architectures used by DMA are DEC products.

8) Productivity:

An increased capability to provide materials or services. DMA's production output requirement is expected to grow rapidly with image processing enhancements.

9) Resource requirements:

The amount of computer resources, including labor, necessary to perform a task. DMA, as with any organization, has limited resources with which to accomplish its assigned task.

9.3 TOOL DEMONSTRATIONS

The month of June 1981 was selected for tool demonstrations/presentations, two weeks at each center.

9.3.1 Tools Presented

Several vendors agreed to give presentations/demonstrations on the following tools:

1) On-Line Systems - OSCAR, OLIVER
2) Univac - RPS, CTS, MAPPER, ASET
3) Interactive Systems, Inc. - IS/1 Workbench for the VAX
4) Systems Engineering Laboratories - SOFTOOL 80
5) General Dynamics Data Systems Division (Eastern Center) - STAR 1100
6) General Dynamics Data Systems Division (Central Center) - PRICE S
7) Logicon - LARE
8) Grumman Aerospace Corporation - SOLID
9) Gilbert Commonwealth - CUE
10) High Order Software - FAME.

GD/DSD DMA project personnel presented all other tools:

- ATA
- AUTOFLOW
- DAVE
- COMPARATOR
- FORMAT
- FORTRAN'77 ANALYZER
- LOGOS
- NODAL
- SCMS
- SDDL
- SFTRAN3
- USER INTERFACE FOR ON LINE ASSISTANCE (UIFOLA)
- UPDATE.

9.3.2 Demonstration Requirements/Schedule

The physical requirements of the presentations were discussed and agreed upon at the Status Review held in St. Louis at DMAAC on 19-20 May 1981 (see Figure 9.1). The demonstrations were held in consecutive two-week periods at DMAHTC and DMAAC respectively (see Figure 9.2). The first two weeks of June were designated for presentations at DMAHTC with duplicate presentations at DMAAC the following two weeks. Those tools outlined in cross-hatching were presented by their vendor; all other tools were presented by GD/DSD personnel. After each presentation a survey form was completed by the DMA attendees to evaluate the tool with respect to applicability and appropriateness to DMA needs. The number of personnel responding by tool and center is listed in Figure 9.3. A copy of this survey form is included as Appendix B.

The differences in the schedule between the two centers developed from facilities and manpower arrangements. At DMAHTC the conference room available was small and a select group was chosen to participate in all presentations. The limited seating required multiple presentations in most cases to allow for maximum dissemination of information. At DMAAC the conference room available was much larger leading to a reduction in the number of double presentations.

- FCR TWO WEEK PERIOD AT EACH CENTER
- ATTENDEES EXPECTED: 5 -10
- STARTING TIMES
  - MORNING SESSION: 9:30 a.m.
  - AFTERNOON SESSION: 1:00 p.m.
- EXPECTED TIME REQUIREMENT: 1 - 2 HOURS PER SESSION
- PHYSICAL REQUIREMENTS
  - CONFERENCE ROOM WITH TELEPHONE LINE
  - OVERHEAD PROJECTOR
  - 35mm SLIDE PROJECTOR
  - CRT AND MODEM
  - PRINTER (132 column capability)
  - VIDEOPLAYER (optional)

Figure 9.1 TOOL PRESENTATIONS AND DEMONSTRATIONS

	MON	TUE	WED	THU	FRI	
at DMAHTC	1 AUTOFLOW UPDATE	2 FORMAT SFTRAN3 LOGOS	3 UIFOLA DAVE FTRAN'77 ANA	4 PRICE "S"*	5 STAR-1100*	9:30
	SDDL SCMS		////// /FAME/ //////	ATA COMPARATOR NODAL	/////// /LARE// ///////	1:00
	8 ///////// /OLIVER// /OSCAR/// /////////	9 //////////// /SOFTOOL 80/ ////////////	10/////////// /IS/1////// /WORKBENCH/ /FOR VAX/// ///////////	11 /////// /SOLID/ ///////	12///////// /1100//// /INTER-// /ACTIVE// /////////	9:30
	///////// /OLIVER// /OSCAR///	////////////// /SOFTOOL 80/// //////////////	///// /CUE/ /////		///////// /MAPPER// /////////	1:00
at DMAAC	15 ATA COMPARATOR NODAL	16//////// /1100/// /INTER-/ /ACTIVE/ ////////	17 UIFOLA DAVE FTRAN'77 ANA	18 /////// /SOLID/ ///////	19 STAR-1100	9:30
	SDDL SCMS	//////// /MAPPER/ ////////	///// /CUE/ /////		////// /LARE/ //////	1:00
	22 //////// /OLIVER/ ////////	23 //////////// /SOFTOOL 80/ ////////////	24/////////// /IS/1////// /WORKBENCH/ /FOR VAX/// ///////////	25 AUTOFLOW UPDATE	26 ////// /FAME/ //////	9:30
	/////// /OSCAR/ ///////	////////////// /SOFTOOL 80/// //////////////	FORMAT SFTRAN3 LOGOS	PRICE "S"		1:00

Figure 9.2 Presentation/Demonstration Schedule

	DMAHTC	DMAAC
ATA:	5	12
COMPARATOR:	5	10
NODAL:	6	11
MAPPER:	4	6
UIFOLA:	3	12
DAVE:	3	10
FORTRAN 77 ANALYZER:	4	10
SOLID:	6	6
STAR:	7	5
SDDL:	8	10
SCMS:	6	10
INTERACTIVE 1100:	5	11
CUE:	3	7
LARE:	4	5
OLIVER:	7	5
OSCAR:	5	7
SOFTOOL 80:	3	16
IS/1 WORKBENCH:	3	9
AUTOFLOW:	6	9
UPDATE:	6	9
FORMAT:	7	8
SFTRAN3:	7	9
LOGOS:	7	8
FAME:	4	11
PRICE S:	9	9

Figure 9.3 RESPONSES BY TOOL FROM DEMONSTRATION

9.4 DEMONSTRATION RESULTS

Most of the DMA personnel involved in the demonstrations did not have experience using software tools in all phases of the life cycle. As a result, the data gathered was most useful as a means of determining their understanding of the presentations, but was not heavily weighted in selecting the tools to be evaluated in phase II.

9.4.1 Analytical Method

The results of the presentations were statistically analyzed through the tabulation of questions on evaluation sheets distributed by GD/DSD. The data was then reduced through several steps to two lists from DMA responses: "most

positive" and "least negative". The ranking of the tools was based upon the number of survey responses. Survey questions used for ranking were based upon a tool's applicability to DMA tasks and ease of use in an interactive environment. The first step in this process involved gathering statistics from the sheets. The responses were analyzed for general responses to get a perception of how well the presentations were understood and whether or not the tools would be acceptable in the DMA environment.

Next, the data was sorted by tool and then by DMA center. Seven questions were selected to be used in generating numerical data to support the general responses. These questions are annotated in Figure 9.4. The demonstration response form is included as Appendix B. The statistics associated with these questions are displayed in Figures 9.5 and 9.6 for DMAHTC and DMAAC, respectively. Since the objective was to develop a ranking of the tools by positive and negative associations, the questions had to be assigned a corresponding connotation.

For questions, Q1 and Q3 (see Figure 9.4), a rating of "high" was considered positive and "low" negative, since this implied ease of use. Questions Q5 and Q6 were considered to be positive with a "yes" answer, the implication being an application of the tool could be perceived in the DMA environment. An indication that modifications would be required to a tool in order to make it easier to use were considered as negative responses.

This rationale lead to using a "no" response to questions Q2 and Q4 as a "positive" answer.

Finally, question Q7 was considered to have a negative implication. This decision was made based upon knowledge derived from prior activities and discussions. While the fact that a tool performs similar functions to tools available at DMA is not in itself negative; a consideration must be given to the fact that the tools at DMA are not being utilized, for numerous reasons. The association of a tool being presented to a tool "available", but <u>not</u> considered "useful", would be considered negative.

Questions which were not responded to were not considered in the tabulations. Also, answers of "medium" for questions Q1 and Q3 were classified as neutral responses in the tabulations.

9.4.2 Summary Data

The total number of responses by question and center are given in Figure 9.7, along with summary data. Again, the lower number at DMAHTC can be attributed to the approach taken in participating in the presentations by the center and does not reflect on the quality of the statistics. The trend responses give an indication of the acceptance to possible changes in the software tools available. A positive attitude on the part of the participants is evident, though not conclusive.

A summary of all of the statistics is given in Figure 9.8. This summary breaks-out the data by tool, character of response and center on the left side of the chart. Numeric totals are then collected in the middle of the chart; first by character of response and then without consideration of any class grouping. Using the last total the data is normalized in the three right-most columns by "positive", "negative" and "neutral" connotations.

9.4.3 Tool Rankings

Using the positive and negative columns of normalized data the rankings of tools by DMA in Figure 9.9 were derived. The higher a tool appears in the "most positive" list, the larger the number of positive comments received. The capabilities of some demonstrated tools may have been new to the evaluators, and hence they did not immediately perceive a use for that tool. Therefore, in order to provide a uniform baseline for ranking familiar and unfamiliar tool capabilities, the DMA survey responses were also ranked by the least number of negative comments recorded. The middle column lists in order the tools based upon the number of <u>negative</u> survey responses. Again, the best tool is at the top. The lower a tool appears in the list the larger the number of negative comments received.

A line was drawn connecting the same tool in the most positive and least negative rankings. The more horizontal the slope of this line the higher the correlation between the two ranking schemes. A line with a very steep slope indicates an uncertain correlation. Such tools were probably not understood, or an application was not apparent. The most desirable tools are those that are near the top of both lists and have a high correlation indicated by a nearly horizontal line.

A third list, "most positive", was also generated by DSD/Central Center project personnel for comparison with DMA responses (see Figure 9.9). The tool ranking was based on the needs of the Defense Mapping Agency as perceived by the

project team members from their experience on the task. The higher a tool appears in the list, the more applicable to the DMA environment. Those tools that appeared at the top of the three lists and had consistent correlations were desirable candidate tools for the phase II evaluation. As explained in Section 9.0, this was only one of several criteria used to choose the phase II tools.

(GENERAL EXPLANATION OF TERMS)

QUESTIONS USED TO GATHER STATISTICS*

Q1: Your evaluation of the ease of input data preparation:
_____ High, _____ Medium, _____ Low

Q2:** Are there modifications to the input data preparation that would make the tool easier to use in the DMA environment? _____ Yes, _____ No

Q3: Your evaluation of the ease of understanding the output results:_____ High, _____ Medium, _____ Low

Q4:** Are there modifications to the output results format that would make the tool more useful in the DMA environment? _____ Yes, _____ No

Q5: Do you perceive an application of the tool to DMA projectsin the near-term (FY 1982)? _____ Yes, _____ No

Q6: Do you perceive an application of this tool to DMA projects in the far-term (FY 1985)? _____ Yes, _____ No

Q7:** Does this tool have functions that are also present in currently available DMA tools? _____ Yes, _____ No

H - High
M - Medium
L - Low

Y - Yes
N - No

*Not all questions were answered on all questionnaires.
**Negative Implications

Figure 9.4 DSD/DMA TOOL DEMONSTRATION

	Q1			Q2		Q3			Q4		Q5		Q6		Q7	
	H	M	L	Y	N*	H	M	L	Y	N*	Y	N	Y	N	Y	N*
ATA	0	3	1	1	1	0	3	0	0	2	2	2	2	0	1	3
COMPARATOR	1	2	0	2	2	3	0	0	0	3	4	1	3	1	2	2
NODAL	0	3	0	0	3	0	4	0	0	4	2	4	4	2	2	3
MAPPER	4	0	0	0	4	4	0	0	0	3	2	2	1	1	0	4
UIFOLA	1	0	1	1	0	1	0	1	0	1	1	2	2	1	0	2
DAVE	2	0	0	1	1	1	2	0	0	1	1	2	2	0	2	1
FORTRAN 77 ANALYZER	1	2	1	0	1	0	2	0	0	1	1	2	3	0	1	1
SOLID	0	3	1	2	3	0	4	1	1	3	1	5	4	0	2	1
STAR	4	2	0	0	6	3	4	0	1	5	5	2	4	1	1	4
SDDL	1	5	2	2	6	3	4	0	1	6	3	4	7	1	4	4
SCMS	0	3	1	0	3	0	2	1	0	3	2	3	5	1	0	6
INTERACTIVE1100	3	1	1	1	3	4	1	0	0	5	4	1	3	0	4	1
CUE	1	2	0	0	3	1	2	0	0	3	0	3	2	0	0	1
LARE	0	1	3	0	1	0	2	1	0	2	0	4	1	1	0	3
OLIVER	3	3	0	1	6	3	4	0	0	6	4	3	4	1	2	2
OSCAR	1	4	0	1	3	1	4	0	0	3	0	3	2	1	2	1
SOFTOOL 80	2	1	0	0	2	2	1	0	0	2	2	0	3	0	2	0
IS/1 WORKBENCH	0	1	1	1	2	0	1	1	1	1	0	3	2	1	2	0
AUTOFLOW	4	1	1	0	5	5	1	0	3	3	5	1	3	2	4	2
UPDATE	3	3	0	2	3	3	2	0	0	4	4	2	4	2	5	1
FORMAT	3	4	0	0	7	4	3	0	1	6	3	4	4	2	5	2
SFTRAN3	1	5	1	0	5	1	4	0	0	4	3	4	4	3	3	3
LOGOS	3	3	0	0	5	3	2	0	1	4	2	5	3	4	3	3
FAME	0	3	1	1	2	1	3	0	0	2	0	4	3	0	0	1
PRICE	2	7	0	1	8	4	5	0	1	8	4	5	4	3	0	6

*POSITIVE RESPONSE

Figure 9.5 Tool Demonstration Responses - DMAHTC

	Q1			Q2		Q3			Q4		Q5		Q6		Q7	
	H	M	L	Y	N*	H	M	L	Y	N*	Y	N	Y	N	Y	N*
ATA	5	3	0	5	3	6	1	0	1	5	6	3	4	4	6	2
COMPARATOR	4	3	1	4	3	5	1	0	0	6	3	5	2	4	2	2
NODAL	7	1	0	2	4	6	1	0	2	5	6	3	4	2	4	1
MAPPER	4	0	0	0	5	5	0	0	0	5	2	3	3	0	1	2
UIFOLA	5	3	0	0	7	6	1	0	0	7	8	2	8	0	0	5
DAVE	5	2	0	3	5	5	3	0	0	5	6	2	5	2	4	1
FORTRAN 77 ANALYZER	3	4	0	1	4	2	5	0	0	5	7	3	7	2	5	0
SOLID	0	3	0	1	3	1	3	0	1	3	3	2	2	1	0	1
STAR	1	2	0	0	2	2	1	0	0	2	0	1	0	0	0	1
SDDL	2	2	3	0	6	2	4	1	1	4	5	3	4	3	1	2
SCMS	4	3	0	1	5	3	3	1	0	4	5	4	6	3	3	3
INTERACTIVE1100	7	3	0	1	5	7	3	0	0	6	6	3	5	2	2	1
CUE	1	3	0	1	2	1	3	0	1	2	0	4	0	4	1	2
LARE	0	2	0	0	2	0	2	0	0	1	1	2	1	1	0	0
OLIVER	3	1	0	2	3	2	2	0	0	3	1	3	1	2	1	1
OSCAR	2	4	0	0	4	2	4	0	0	4	3	3	2	2	1	2
SOFTOOL	10	3	1	5	6	8	6	0	1	6	10	4	10	1	6	2
IS/1 WORKBENCH	4	4	0	0	5	2	6	0	0	4	4	4	3	3	2	3
AUTOFLOW	2	6	1	5	4	5	4	0	2	6	5	3	4	2	4	2
UPDATE	2	3	3	2	6	4	4	0	1	7	0	7	0	8	3	1
FORMAT	1	6	1	3	5	6	1	0	1	6	1	6	1	6	5	0
SFTRAN3	4	4	0	0	7	3	4	0	0	6	5	3	2	4	3	4
LOGOS	5	3	0	0	7	4	4	0	0	7	0	6	0	6	5	1
FAME	2	4	2	4	3	1	5	2	2	4	2	5	3	3	0	5
PRICE	4	3	0	4	3	5	2	0	3	3	6	1	3	1	0	4

*POSITIVE RESPONSES

Figure 9.6 Tool Demonstration Responses - DMAAC

TOTAL RESPONSES

	Q1			Q2		Q3			Q4		Q5		Q6		Q7	
	H	M	L	Y	N*	H	M	L	Y	N*	Y	N	Y	N	Y	N*
HTC	40	62	15	17	85	47	61	5	10	85	56	71	79	28	47	57
AC	88	75	12	44	109	93	72	4	16	116	95	85	80	66	59	48
TOTAL	128	137	27	61	194	140	133	9	26	201	151	156	159	94	106	105

	HTC	AC
RESPONSES	133	225

TREND RESPONSES

	H&Y	M	L&N
HTC	449	123	193
AC	629	147	286
TOTALS	1078	270	479

* POSITIVE RESPONSE

Figure 9.7 Total Responses

	POSITIVE		NEUTRAL		NEGATIVE		TOTAL			SUM	% POSITIVE	% NEUTRAL	% NEGATIVE
	HTC	AC	HTC	AC	HTC	AC	P	N	N				
ATA	10	31	6	4	5	19	41	10	24	75	55	13	32
COMPARATOR	18	25	2	4	6	16	43	6	22	71	61	8	31
MODAL	16	33	7	2	8	13	49	9	21	79	62	11	27
MAPPER	22	26	0	0	3	4	48	0	7	55	87	0	13
UIFOLA	8	46	0	4	6	2	54	4	8	66	82	6	12
DAVE	9	32	2	5	5	11	41	7	16	64	64	11	25
FORTRAN 77 ANALYZER	8	28	4	9	4	11	36	13	15	64	56	21	23
SOLID	12	13	7	6	12	5	25	13	17	55	45	24	31
STAR	31	8	6	3	5	1	39	9	6	54	72	17	11
SDDL	30	25	9	6	14	12	55	15	26	96	57	16	27
SCMS	19	30	5	6	6	12	49	11	18	78	63	14	23
INTERACTIVE1100	23	37	2	6	7	8	60	8	15	83	72	10	18
CUE	11	8	4	6	3	11	19	10	14	43	44	23	33
LARE	7	5	3	4	9	3	12	7	12	31	39	22	39
OLIVER	28	14	7	3	7	8	42	10	15	67	63	15	22
OSCAR	11	19	8	8	7	6	30	16	13	59	51	27	22
SOFTOOL 80	13	52	2	9	2	18	65	11	20	96	68	11	21
IS/1 WORKBENCH	5	25	2	10	10	9	30	12	19	61	49	20	31
AUTOFLOW	27	28	2	10	11	17	55	12	28	95	58	13	29
UPDATE	22	20	5	7	11	24	42	12	35	89	47	14	39
FORMAT	29	20	7	7	12	22	49	14	34	97	51	14	35
SFTRAN3	21	31	9	8	11	10	52	17	21	90	58	19	23
LOGOS	23	24	5	7	13	17	47	12	30	89	53	13	34
FAME	9	20	6	9	6	18	29	15	24	68	43	22	35
PRICE S	36	28	12	5	10	9	64	17	19	100	64	17	19

Figure 9.8 Tool Demonstration Evaluation Summary Statistics

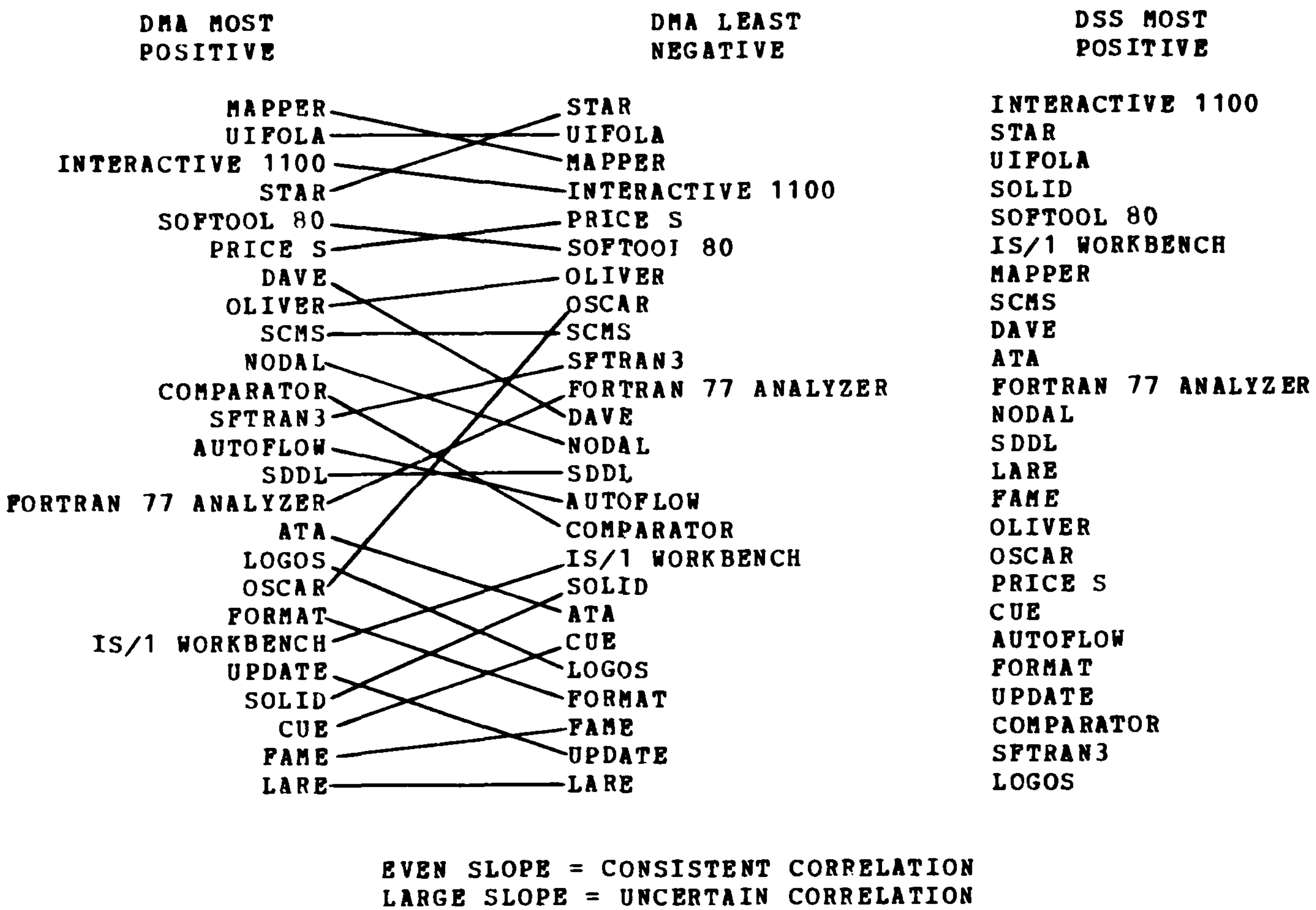


Figure 9.9 Tool Ranking from Demonstrations

10.0 PHASE II ACTIVITIES

In this phase of the project, an in-depth analysis of the DMA software development environment was conducted through the simulation of an actual development scenario using selected tools. The tools being evaluated were representative of a larger class of software tools. The availability of training, technical support and computer resources were the underlying considerations in the selection process. The data from the demonstrations was also used as a selection mechanism. These constraints provided for an in-depth, informative evaluation phase.

10.1 EVALUATION ACTIVITY AND TOOL SELECTION

Figure 10.1 summarizes the rationale behind the selection of the tools for the evaluation activity. These tools were to be considered representative in nature of all the tools under consideration for the DMA MPE. Data was gathered about how the tool could be applied at DMA, the shortcomings of its use, the characteristics which seemed most productive, and methods and training required to introduce the tool into the programming environment. These data were applied as constraints against all tools considered in a specific life cycle activity. The implementation of this task is described in Section 11.0. Figure 10.2 presents the tools which were demonstrated in June, 1981 and not selected for the evaluation phase. The rationale of the selection/non-selection process is detailed in Sections 10.1.1 and 10.1.2 respectively.

An optimum evaluation tool set was chosen on the basis of available funding, potential benefit, near-term applicability to DMA, vendor support, and available computer/manpower resources. In addition, tools had to be selected which would cover all the phases of the software development life cycle as well as support associated management activities as defined in the SON. Descriptions of the tools selected for the tool evaluation have been included as Appendix C. Due to evaluation constraints, as described in Section 10.1.5, only two development scenarios were followed, one at each center. The development scenario was unique to each center; however, the test-bed problem was the same. Therefore, in addition to evaluating the individual tool capabilities, it was possible to evaluate the different development scenarios. GD/DSD was to develop the program independently as a third scenario for comparison with the centers. This was only accomplished through the design phase. Independent development by GD/DSD was stopped when it was assessed that both centers would

finish all life cycle tasks for the test-bed problem within the specified schedule.

Commonalities between the two center's planned scenarios included the same requirements, testing and project management tools; Front-end Analysis and Modeling Environment (FAME), Software Complexity Measurement System (SCMS), OPTIMA and MAPPER respectively. Each center developed the requirements for the system using the FAME tool independently, and those requirements were used as the input into the design phase. The design at DMAHTC was accomplished through the expanded use of FAME while at DMAAC SDDL was utilized. DMAHTC used an integrated software system for coding and source documentation, Interactive Systems/One (IS/1) Workbench, while DMAAC used TX. FORMAT was to be evaluated at DMAAC, but access time was limited due to hardware problems; and only a cursory review was accomplished. NODAL was used as a testing tool for both centers' programs. At DMAAC team members were shown at the Harris terminal how to set up a NODAL run for their program files and output from a sample run. At DMAHTC, due to line problems, a live demonstration could not be given. Instead, sample jobstreams and outputs were shown, explained and discussed for a small sample problem with which they were familiar. GD/DSD later ran the DMAHTC source through NODAL. SCMS was to be used but the source code was not obtained from the supporting government agency within the evaluation period. It was preferred because of metrics provided as part of its output. By using the same tool to evaluate the testing phase of both development scenarios a better evaluation of the design and coding phase tools was achieved. This did not apply to the requirements phase since both scenarios used the same tool. A Digital Land Mass System was implemented in both scenarios, though the techniques and methods were different. The coding at each center was accomplished in FORTRAN'66 (FORTRAN IV). This level of FORTRAN was dictated by the availability of mature testing tools.

The project management system, OPTIMA, was not utilized. OPTIMA, a software program developed and hosted on a UNIVAC, was not available in an exercisable format. However, an interactive presentation was available on the UNIVAC Demonstration and Presentation Computer System (DAPS) in Eagan, Minn. through a remote dial-up link. An additional management tool exercised was a data base inquiry/report generator system, MAPPER, used in collecting data during the evaluation as described in Section 12. Access was accomplished through terminals and modems supplied by UNIVAC to their DAPS.

10.1.1 Selected Tools

The requirements tool, FAME, was used to specify the requirements of the test-bed program. This tool was selected after discussions with Higher Order Software (HOS), FAME's vendor. HOS offered more support and better computer resource utilization rates than other vendors. This tool was also used to specify the design at DMAHTC. SDDL (Software Design and Documentation Language), developed at the Jet Propulsion Laboratories, was used for the design phase at DMAAC. GD/DSD had experience with this tool and found it to be user-friendly and very productive. The computer resource costs associated with the tool are low because it is hosted on a minicomputer and it is highly portable, having been written in Jenson and Wirth PASCAL. At DMAHTC a UNIX based system called "IS/1 Workbench for the VAX" was utilized during the coding and documentation phases. This system was selected for use at DMAHTC because of the availability of support. Interactive Systems, Inc., the vendor, has a Washington D.C. office. A high rating during the June demonstrations and good vendor support in providing documentation, terminals and training, were other reasons this integrated system was selected. FORMAT and TX were to be used for the documentation and coding, respectively, at DMAAC. Both are hosted on the same system as SDDL, a low cost system with good vendor support. Unfortunately, as previously stated, FORMAT was not thoroughly evaluated. The testing tool used by GD/DSD was NODAL. This tool had fair ratings in the demonstrations and is in the public domain. This tool was hosted at DSD/Central Center on the SES system providing for low resource costs. MAPPER received the highest ratings during the presentations. The UNIVAC support was very good and there was historical data on MAPPER's use from multiple companies. The resource costs associated with this system were much lower than those of other systems available during the tool evaluation period. OPTIMA was chosen as the project management tool for similar reasons. It was already hosted on UNIVAC equipment and could be used at low cost. This set of tools kept the number of vendor contacts, computer systems and training trips required for the evaluation task to a minimum, while fulfilling the requirement of effectively evaluating the DMA programming environment and of exercising life cycle support tools. Figure 10.1 provides a summary of this information.

10.1.2 Non-Selected Tools

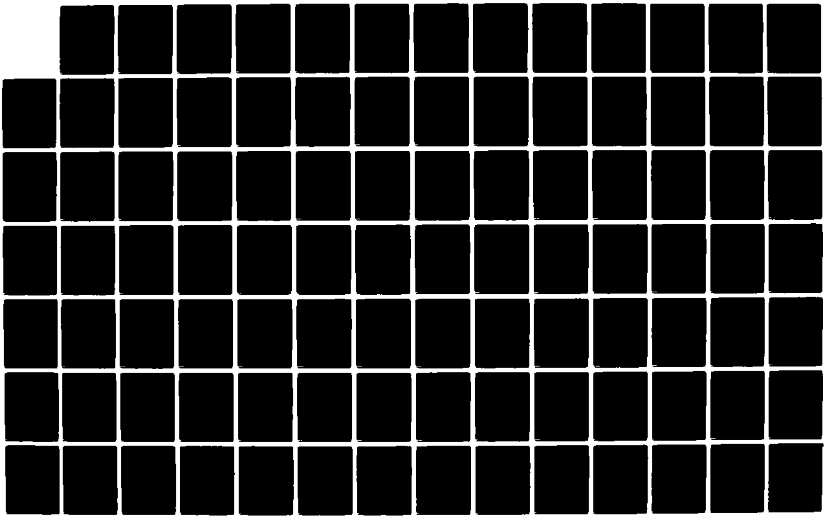
The main factor in not selecting tools was their inability to fit into the limited scenarios of the evaluation. The fewer hardware/software systems which could be used the less com-

plex the plan. If a tool was not considered a high need or was not easily accessed, it was not considered in the plan, unless it was required to cover part of the life cycle development. The only tool falling into this category was FAME.

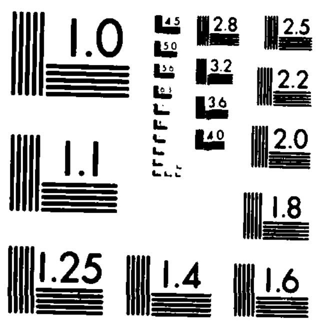
AD-A125 626 INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY VOLUME 1(U) GENERAL DYNAMICS FORT WORTH TX FORT WORTH DIV H C CONN ET AL. JAN 83 DMA-2-014-VOL-1

2/5

UNCLASSIFIED RADC-TR-83-3-VOL-1 F30602-81-C-0039 F/G 9/2 NL



M-2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

10.1.3 Test-Bed Program

The sample computer program to be developed at each center was drawn from the cartographic application area and was modeled after the Digital Land Mass System (DLMS) problem used in a software engineering training class conducted at DMAAC. The problem was to design a new production program which would manipulate a cartographic data base in digital form. The data base consisted of one file containing four manuscripts, which could possibly overlap. Each manuscript contained point, linear and areal features in coordinate form along with associated descriptive information. The program requirements included the extraction of data and features, combination and merging of manuscripts, updating the features of a manuscript, and verification of interdependencies of feature types. The number of features per manuscript was minimized to the extent that all tasks might be accomplished. The entire program was expected to be between 200 and 400 lines of FORTRAN'66 code. This was an estimate and was not to be considered a goal. If the time constraint did not allow for the development of all the requirements, the tasks were to be narrowed in scope to allow completion of all life cycle phases. A maintenance update activity would also have been undertaken, however, at both centers the scope had to be narrowed because of time limitations.

10.1.4 DMAAC Scenario

In the following two paragraphs an explanation of the scenarios with respect to the support tools and their interfaces is provided. The first paragraph provides insight into the original plan while the second provides information associated with the problems encountered.

10.1.4.1 Planned Scenario

The DMAAC planned scenario involved the use of a non-integrated set of tools to define requirements, design, code, test and document the test-bed program. PAME, hosted on a VAX and accessed over a 300 or 1200 baud line, was to be used for the test-bed program requirements specification and SDDL to design the software. SDDL access was specified to be through a modem (300 baud) to the Software Engineering System (SES) at DSD/Central Center over a Harris terminal supplied by GD/DSD. The Harris (SES) full-screen editor (TX) would be used for code entry and the Harris text processor (FORMAT) for documentation using the same terminal. SCMS also hosted on the SES was to be used for program testing, and MAPPER and OPTIMA, hosted on UNIVAC equipment, was to be used for the data base management and project management, respectively.

Hence, all life cycle phases of the test-bed program and automated project management were to be simulated in the DMAAC scenario.

10.1.4.2 Actual Scenario

The DMAAC scenario went as planned with the following exceptions. First, there were power supply problems with the Harris terminal and it was 28 October 1981 before the source of the problem was discovered and a replacement terminal delivered. As a result, a TYMSHARE 126 and Harris' line editor were used for the design phase and the start of the coding phase. Also, as the team was already familiar with the UNIVAC ED processor available on their Univac system, FORMAT was demonstrated but not used. In addition, the UNIVAC UTS 400 terminal to be used with OPTIMA and MAPPER was not received by the team until 10 November 1981. As explained in Section 10.1, OPTIMA was not available in exercisable form. Instead, an interactive presentation was available through the UNIVAC DAPS. This meant that no real evaluation of this project management system was accomplished. MAPPER was available in exercisable form and used by individual team members for storing their evaluation statistics once the terminals arrived. Finally, due to time constraints and implementation problems in the hosting of SCMS on the Harris 500 (SES), NODAL was used instead of SCMS during the testing phase.

10.1.5 DMAHTC Scenario

In the following two paragraphs, an explanation of the scenarios with respect to the support tools and their interfaces is provided. The first paragraph provides insight into the original plan while the second provides information associated with the problems encountered.

10.1.5.1 Planned Scenario

The planned DMAHTC scenario used IS/1 Workbench for VAX, an integrated system similar to UNIX, to code and document the test-bed program. The SHELL, SCCS and MAKE utilities of the system were to be used for coding, and the special word-processing functions for document generation. Access was to be through a 1200 baud modem and INtext terminal supplied by Interactive, Inc to a VAX system in Santa Monica, CA. In the scenario, ‹AME, SCMS, MAPPER and OPTIMA tools were to be used for requirements and design, testing, data base management and project management, respectively, hosted on the systems indicated in Section 10.1.4.1.

10.1.5.2 Actual Scenario

The DMAHTC scenario was implemented as planned with the following notable exceptions. As at DMAAC, an OPTIMA tutorial was available for study but the tool was not used for management of the DLMS project due to a presentation only system being available. Also, as at DMAAC, NODAL instead of SCMS was used during the testing phase by GD/DSD. In addition, the SHELL command language and MAKE utility, both part of the IS/1 Workbench for the VAX, were not used during the coding phase as planned. These tools, though they can be used with FORTRAN programs, were written to be used with the C language. SHELL did not recognize the VAX FORTRAN command, so command files could not be set up to do compiles and related functions. Also, though told MAKE could be used with FORTRAN programs to control files, it apparently takes some special handling and could not be made to work during the period allotted to the coding phase. Instead, the LIBRARY command available on the VAX was used to create object module libraries which worked quite well for the DLMS test-bed program. SCCS, Source Code Control System, was demonstrated and analyzed using small sample demonstration programs, but as time did not allow a maintenance effort, this configuration control system was not used for the DLMS program.

SELECTED	CENTER	PHASE SUPPORTED	REASON
FAME	AC&HTC HTC	REQUIREMENTS DESIGN	LOWEST COMPUTER COST VENDOR SUPPORT
SDDL	AC	DESIGN	DSD EXPERIENCE LOW COMPUTER COST
TX	AC	CODING	VENDOR SUPPORT LOW COMPUTER COST
IS/1 WORKBENCH FOR VAX	HTC HTC	CODING DOCUMENTATION	HIGH DEMO RATING VENDOR SUPPORT
FORMAT	AC	DOCUMENTATION	LOW COMPUTER COST VENDOR SUPPORT
SCMS	AC&HTC	TESTING	DMA ENVIRONMENT - FORTRAN'66 OPERATES UNDER UNIX APPLICABLE TO STRUCTURED FORTRAN & PASCAL
MAPPER	AC&HTC	DATA RETRIEVAL	DMA ENVIRONMENT - UNIVAC VENDOR SUPPORT
OPTIMA*	AC&HTC	PROJECT MANAGEMENT	DMA ENVIRONMENT - UNIVAC VENDOR SUPPORT

*NOT DEMONSTRATED IN JUNE

Figure 10.1 RATIONALE FOR SELECTED TOOLS

NON-SELECTED	REASON
ASET**	ALREADY CONSIDERED FOR PURCHASE BY DMA
RPS**	NOT APPLICABLE TO DMA ENVIRONMENT - USER COMMUNITY
CTS**	TEP+CONSTRAINTS - ONLY TWO SCENARIOS
LARE	EXCESSIVE COSTS FOR EVALUATION
NODAL & ATA	REDUNDANT TO FORTRAN'77 ANALYZER TEP CONSTRAINTS - NOT USING FORTRAN'66 IN SCENARIOS
OLIVER	REDUNDANT TO MAPPER EXPENSIVE TO EVALUATE DMA ENVIRONMENT - NOT UNIVAC
OSCAR & CUE	EXPENSIVE TO EVALUATE DMA ENVIRONMENT - NOT UNIVAC REDUNDANT TO OPTIMA
PRICE S	CONSTRAINTS OF TEP - COST
SOFTOOL 80	TEP CONSTRAINTS - INSUFFICIENT PERSONNEL
AUTOFLOW & LOGOS & COMPARATOR	NO NEED IDENTIFIED AT DMA
DAVE & UPDATE	NOT HOSTABLE - CDC SYSTEM ONLY HIGH UTILIZATION COSTS
FORTRAN'77 ANALYZER	NOT RELEASABLE - ERRORS
SFTRAN3	NOT RATED HIGH DURING DEMO'S SIMILAR TO ASCII FORTRAN
SOLID & UIFOLA	NOT AVAILABLE FROM DEVELOPER
STAR 1100	SYSTEM TOOL - NOT PART OF TEP SCENARIO
FAVS	ALREADY BEING EVALUATED BY DMA

** PARTS OF "INTERACTIVE 1100" PRESENTATION
+ TOOL EVALUATION PLAN

Figure 10.2 Rationale for Non-Selected Tools

10.2 GD/DSD TOOL TRAINING

Prior to the start of the phase II on-site tool evaluations, DSD/Central Center personnel received training and practice in the use of the tools selected. During 24-25 August 1981, training was received in the use of the IS/1 Workbench from Interactive Systems, Inc. On 14-15 September 1981 training was received on FAME from HOS. Then on 1 October 1981, training was provided by UNIVAC on OPTIMA and MAPPER. Additional formal training was provided on the IS/1 system on 1 October with continuous training support being provided through an InterActive, Inc. supplied terminal.

10.3 DMA TRAINING

DSD/Central Center personnel provided training to DMA personnel on each tool to be evaluated prior to the start of its associated life cycle phase on the DLMS test-bed project. The training consisted of an explanation and demonstration of how to access and use the tool, objectives of the hands-on evaluation, summary of the capabilities of the tool, benefits to be derived from the use of the tool, sample scenario to follow during the evaluation, and an explanation of data to be collected by participating DMA personnel.

10.4 SCHEDULE OF ACTIVITIES

Resource constraints, money, time and manpower were some of the factors considered in establishing the phase II tool evaluation plan. At the August 1981 In-Process-Review (IPR), DMA indicated an availability of five technical and two managerial personnel at a 50% level-of-effort over an eight week period. Allowing for a five week time span after the IPR to establish vendor coordination, generate a software development scenario, provide training for project team members, refine the Tool Evaluation Plan (TEP), update the Statement of Operation Need and System Operational Concept (SON/SOC), and have a review in St. Louis, MO, a start date of 05 October 1981 was selected.

A schedule of activities and on-site support, as accomplished during the seven week evaluation period, is presented in Figure 10.3. The period was shortened from eight weeks to seven to provide for maximum continuity of tasks as well as due to holiday factors in the calendar period. On the 5th of October, R. Bond and M. Goode arrived at DMAHTC, Washington D.C. to start the tool evaluation task. The first day an overview of the effort was presented, including tools to be utilized, access methods, schedule of activities, and tasks to be accomplished as described in Section 10.3. Then

specific tasks were assigned. All personnel were to work on each life cycle task, but a different person would be assigned the lead role during each life cycle phase. All documentation was delivered at this time so that individuals responsible for leading specific tasks (each life cycle phase was a task) could become familiar with the associated tool. The next two days entailed performing the requirements task. The last day was used to start the design phase. From 13-16 October, a duplicate scenario was conducted at DMAAC, St. Louis by R. Bond. Each center was responsible for continuing the efforts in the assigned tasks when GD/DSD departed.

Next, it was planned that on the 22nd of October, M. Goode would arrive in Washington and R. Bond in St. Louis for two days to start the evaluation of the project management tools. Up to this point in time these tasks were to have been accomplished manually. However, as explained in Sections 10.1.4.2 and 10.1.5.2, the UTS 400 terminals to be used with the management tools OPTIMA and MAPPER were late in arriving. The trips were still made as assistance trips. At DMAAC, the design phase was continued using the Software Design and Documentation Language (SDDL). At DMAHTC the design phase was completed using the Front-end Analysis and Modeling Environment (FAME) tool and coding was begun using the INed editor which is part of the IS/1 Workbench for the VAX. An extra trip was made to DMAAC on 28 October to deliver a Harris SES terminal and initiate the coding phase with introduction of TX, the Harris' full screen editor. FORMAT, Harris' text processor, was also demonstrated. Simultaneous trips to each center were made again on 02-04 November when the coding phase was continued and testing begun. During the DMAAC visit, the evaluation team went to the local Sperry-Univac office where the data base management system MAPPER was demonstrated and report ID's set-up for each team member to be used in the collection of evaluation statistics. At DMAHTC, MAPPER was introduced and IS/1's Source Code Control System (SCCS) was demonstrated. Next, on 09 November a trip was made to DMAAC and on 10 November to DMAHTC to deliver modems to be used with Univac UTS-400 terminals in accessing MAPPER. At DMAHTC MAPPER was demonstrated and reports set-up for each evaluation team member. These reports were used in the same manner as at DMAAC for the collection and storage of evaluation statistics. Finally, on 16-20 November, R. Bond and M. Goode traveled to both centers for two days each. On the first day the use of NODAL was evaluated. The following day the management tool, MAPPER, was utilized to assimilate the statistics which were gathered during the evaluations, and discussions were conducted between GD/DSD and DMA team members on all aspects of the evaluation effort. These included the individual tools and their features, the inte-

gration aspects of using the tools to support the various life cycle phases, and the usefullness of the test-bed problem in learning about the tools and their capabilities as they applied to the DMA environment.

Statistics gathered from the test-bed development project show a 49% level-of-effort by both centers. DMAAC had six personnel assigned to the task, and DMAHTC seven. The centers had 25 and 30 working days available during the evaluation period, respectively. DMAAC expended 588 hours, an average of 98 per person; and DMAHTC expended 828 hours, an average of 119 per person; for a total of 1416 hours. This does not include time involved in the discussions conducted during the last week of the evaluation.

10.5 GD/DSD OFF-SITE EVALUATION ASSISTANCE

During those times when GD/DSD project personnel were not on site at a DMA center, any questions about the use of a tool, the test-bed problem, etc were directed to GD/DSD by several means. First, telephone numbers for contact were provided. Secondly, the DMAHTC team, while using the IS/1 Workbench, had an interactive mail system (INmail) available to them through which they could communicate with GD/DSD project team members in Fort Worth, who also had an IS/1 terminal. Thirdly, for the DMAAC team while using the Harris SES system located in Fort Worth, special message files were set-up for communication with GD/DSD project members.

```
.(--SEP--) (--------------------------OCTOBER-------------------------).
.M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS.
.28 29 30 1  2     5  6  7  8  9     12 13 14 15 16    19 20 21 22 23    .
.SSSSSS            WWWWWWWWWWWW      SSSSSSSSSSSS                SSSSSS  .R. BOND
.(STATUS           WWWWWWWWWWWW                                  WWWWWW  .M. GOODE
. REVIEW)          #----------#      #----------#                #----#  .
.                                                             HHHBBBBBB  .PROJECT MANAGEMENT
.                  HHH               AAA                                 .INTRODUCTION
.                    HHHHHHH           AAAAAAA                           .REQUIREMENTS
.                          HHHHHH    HHHHHHHHHHBBB   AAAAAAAAAAAAAAA     .DESIGN
.                                                    HHHHHHHHHHHHHHH     .CODING
.                                                                        .
.(---OCTOBER----) (---------------------NOVEMBER----------------------).
.M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS_M__T__W__T__F__SS.
.26 27 28 29 30    2  3  4  5  6     9  10 11 12 13    16 17 18 19 20    .
.                  SSSSSSSSS                           WWWWWWSSSSSSSSS   .R. BOND
.     SSS          WWWWWWWWW      SSSWWW               WWWWWWSSSSSSSSS   .M. GOODE
.                  #-------#                           #-------------#   .
.                                                                        .DESIGN
.BBBBBBBBBBBBBBB   BBBBBB                                                .CODING
.                       BBBBBBBBBB    BBBB   BBBBBB                      .TESTING
.                                                     HHHHHHHAAAAAAAA    .END AND CLOSE
.------------------------------------------------------------------------.

WWWWWWW - AT WASHINGTON D.C.
SSSSSSS - AT ST. LOUIS, MO
#-----# - GD/DSD ON SITE SUPPORT*
HHHHHHH - DMAHTC WORK
AAAAAAA - DMAAC WORK
BBBBBBB - SAME PHASE IN WORK AT BOTH CENTERS
```

Figure 10.3 Tool Evaluation Schedule of Activities (page 1 of 2)

```
.(_________JULY_________)(________AUGUST________)(______SEPTEMBER_______).
.-------------------------------1982---------------------------------------.
.                                                                          .
.                               #__#                                       .INITIAL TRIP BY GD
.                                                                          . TO HOS
.                                                                          .
.                                #_______________________________________#.DEVELOPMENT OF LORAN
.                                                                          . SOLUTION WITH USE.IT
.                                                                          .
.                                                #__#                      .GD & DMA AT HOS
.                                                                          .
.                                                                     #    .VIDEOTAPE OF LORAN
.                                                                          . DEMONSTRATION/PRESENTATION
.                                                                          . DELIVERED
.                                                                          .
.                                                                         #.FORTRAN SOURCE TAPE
.                                                                          . OF LORAN PROBLEM
.                                                                          . DELIVERED TO
.                                                                          . DMAAC & DMAHTC
.                                                                          .
.--------------------------------------------------------------------------.
```

Figure 10.3 Tool Evaluation Schedule of Activities (page 2 of 2)

10.6 MANPOWER UTILIZATION AND PHYSICAL REQUIREMENTS

Figure 10.4 gives the general requirements for facilities and support provided by each organization. Seven people at DMAHTC and six at DMAAC were used at a 49% level-of-effort for a seven week period. One person was managerial and the remaining team members technical. The manager was tasked with the lead role in utilizing the project management and data base tools. He had additional responsibilities in the area of collecting statistics pertaining to the quality, productivity, and ease of use of the tools in each life cycle phase. The complete list of statistics is included as Appendix F. The method of assimilating the data was manual during the first five weeks of the evaluation, followed by the use of MAPPER interactively. Other managerial tasks were to coordinate organizational activities and act as a center interface or contact point through which problems and data could be transferred to GD/DSD personnel and vendors. The lead role for the individual life cycle phase rotated among the technical personnel. Each of the technical personnel was assigned as lead in one of the five phases of development:

1) REQUIREMENTS
2) DESIGN
3) CODING (IMPLEMENTATION)
4) TESTING
5) MAINTENANCE (DOCUMENTATION)

The lead was responsible for being knowledgeable in the use of the tool associated with the assigned task. On the first day of the effort at each center the lead for each phase was chosen. This person was then given the documentation on the tool associated with his life cycle phase role. The lead studied the documentation prior to the time he would act as lead, direct the efforts of the team during the assigned life cycle phase, and serve as the focal point for discussion of the tool's capabilities during the project review held during the last week of the evaluation period. Additionally, each technical person acted under the direction of the other leaders in a team effort to develop the test-bed program.

The physical requirements of each center included a room to support three terminals with modems and a blackboard for presentations by GD/DSD personnel. A printer (132 columns) was also required and helpful to the participating center personnel. One of the three terminals needed was a teletype (TTY) and it was requested to be supplied by DMA at each site. RADC provided the documentation and tools to be used during the testing phase.

DMA

- o PERSONNEL & TIME FOR EVALUATIONS
  - 50% LEVEL-OF-EFFORT
  - 8 WEEK EVALUATION PERIOD
  - 1 MANAGERIAL
  - 5 TECHNICAL
- o TERMINAL/PRINTER & SUPPLIES *
- o PHYSICAL WORK SITE
- o VENDOR ESCORTS AS REQUIRED
- o PHONE LINES/ACCOUNTS AS NECESSARY

DSD/Central Center

- o VENDOR TRAINING/SUPPORT
- o COMPUTER RESOURCES
- o ON-SITE ASSISTANCE/GUIDANCE
- o TERMINAL/PRINTER & SUPPLIES **
- o SUPPORT DOCUMENTATION

RADC

- o GOVERNMENT SUPPORT OBTAINING TOOLS/DOCUMENTATION
  - NODAL
  - SCMS

DMAAC	*	Anderson/Jacobsen CRT
	*	300 baud modem
	*	132 column off line printer
	**	UNIVAC supplied terminal
	**	2400 baud modem
	**	Harris CRT
DMAHTC	*	Anderson/Jacobsen CRT
	*	300 baud modem
	*	132 column off line printer
	**	UNIVAC supplied terminal
	**	2400 baud modem
	**	InterActive supplied terminal
	**	1200 baud modem

Figure 10.4 TOOL EVALUATION SUPPORT REQUIREMENTS

10.7 VENDOR EVALUATION ASSISTANCE

Each vendor provided technical support throughout phase II. DMA personnel were directed to contact GD/DSD project team members if a problem occurred. If necessary, GD/DSD personnel contacted the vendor for resolution of any problems.

11.0 TOOL EVALUATION DOCUMENTATION

Questionnaires were generated by GD/DSD with DMA and RADC support to gather information about the applicability of each generic tool's implementations to the DMA programming environment. Each questionnaire was associated with a specific life cycle development phase. At the conclusion of the evaluation period at each center, DSD/Central Center personnel held discussions with the tool evaluation control groups. The evaluation forms filled out by these groups were used to lead the discussions. These forms covered such important aspects as ease of access, usefulness, and other desirable features of the tools evaluated. Copies of the forms are enclosed as Appendix D. Appendix E contains a summary of the survey responses for each tool evaluation form including comments by team participants.

Following the group discussions, GD/DSD personnel generated a tool summary form, shown in Figure 11.1. This sheet classifies each tool by life cycle phase and important characteristics derived during the evaluation phase, including the evaluation teams' comments and reactions to the tools, which were inputs for determining usefulness to DMA.

Additionally, the task was documentated through the use of MAPPER, a UNIVAC software tool, by gathering information on productivity related activities such as labor hours and computer hours expended, as well as tool performance. The productivity statistics are included in Appendix F.

TOOL NAME	SOFTWARE LIFE CYCLE PHASE(S) SUPPORTED	POSITIVE COMMENTS	NEGATIVE COMMENTS
FRONT END ANALYSIS & MODELING ENVIRONMENT (FAME)	REQUIREMENTS DESIGN	GRAPHIC OUTPUT DATA FLOW ANALYSIS EASY TO USE EXCELLENT DOCUMENTATION PRODUCED	LIMITED TO SMALL PROBLEMS ERRORS IN SOFTWARE DIAGNOSTICS POOR MINIMAL INTERACTIVE I/O CAPABILITY CHANGES HARD TO MAKE DOCUMENTATION
SOFTWARE DESIGN & DOCUMENTING LANGUAGE (SDDL)	DESIGN	FORMATTED OUTPUT MODULE INVOCATION OUTPUT CROSS REFERENCE CAPABILITY	NO PARAMETER CHECKING SPECIAL FLAGS REQUIRED TOO MUCH UP-FRONT INFORMATION REQUIRED
INTERACTIVE SYSTEMS ONE WORKBENCH FOR VAX (IS/1)	DESIGN CODING	GOOD DOCUMENTATION ADVANCED EDIT CAPABILITY DETACHABLE KEYBOARD SUPPORTS MULTIPLE LEVELS OF USERS	DIAGNOSTICS NOT CLEAR LIMITED CURSOR CONTROL FILE ACCESS INCONSISTANT
TEXT EDITOR (TX)	CODING	FULL SCREEN CAPABILITY RECOVERY CAPABILITY DOCUMENTATION	SPECIAL TERMINAL REQUIRED EASY TO LOSE DATA
MODAL	TESTING	ALLOWS DYNAMIC TESTING PUBLIC DOMAIN	I/O OPTIONS TOO RIGID LIMITED OUTPUT INFORMATION
FORMAT	DOCUMENTATION		NOT VERY POWERFUL
OPTIMA	PROJECT MANAGEMENT	UNIVAC	COMPLEX SET UP
MAPPER	DATA BASE	USER DOCUMENTATION UTS 400 EXCELLENT REPORT GENERATION CAPABILITY	SOE POSITIONING COMPLICATED NO AUTO-UPDATE ON EXIT INABILITY TO REPORT ON MULTIPLE REPORT IDENTIFICATION NUMBERS REQUIRES SPECIAL TERMINALS

Figure 11.1 Tool Characteristics Summary Form

12.0 TOOL EVALUATION CONCLUSIONS

During the testing phase of the tool evaluation it was determined that DMAHTC generated 332 lines of FORTRAN code, and DMAAC 365 lines. These figures do not include comment statements. Neither source was extensively tested, and errors are known to have existed. With this taken into consideration, the statistics show DMAHTC produced .40 lines of FORTRAN per hour and DMAAC .62; or 2.5 hours and 1.6 hours per line of FORTRAN respectively. However this variance does not strictly reflect the use of a unique scenario at each center, with respect to design and coding phases. Part of the reason for this difference is that the DMAAC team was more familiar with the stated problem, FORTRAN, and the use of software tools.

Figures 12.1, 12.2 and 12.3 are compilations of evaluation survey responses and activity statistics. Figures 12.1 and 12.2 include a summarization of the evaluation activity statistics by individual team, Figure 12.1, and for the total evaluation effort, Figure 12.2. The activity statistics as compiled on MAPPER by each participant have been included as Appendix F. Figure 12.3 is a compilation of positive and negative responses by programming phase from the life cycle questionnaires over the entire effort, see Appendix E.

Note that the number of yes and no answers for a given question and center may not add up to the number of team members. Two reasons for this are (1) not every member answered every question and (2) ambiguous answers such as "fairly", "somewhat", "maybe" or "so-so" were counted as both a "yes" and a "no". In addition, if a range was specified for an answer requiring a number, the upper limit was used. Also, comments were in some cases paraphrased to express main content and were prefaced with "AC" or "HTC" to denote team source.

An evaluation of the data indicates that the team with previous exposure to software tools, DMAAC, did not find the tools to be as useful as the team with little knowledge of tools, DMAHTC. This is an important fact when considering the type of background the personnel using newly introduced tools should have. A positive attitude about a tool's usefulness will also be important during the transition phases as new tools are introduced. One implication is to introduce new tools by first training less experienced personnel, eventually phasing in everyone as the tools use becomes more widespread. This supports ideas expressed by

DMA personnel when interviews were first conducted in March, 1981.

Further analysis revealed a positive trend with respect to exposure time to a tool. For example, NODAL had the least use at each center, and an accompanying lowest rating. At DMAHTC due to line problems NODAL was discussed and sample output listings explained but not used. At DMAAC time allowed the set-up and processing of 1 run of their program through NODAL. SDDL and FAME were extensively used at DMAAC and DMAHTC respectively; and these tools evidenced a more positive response. This relationship was corroborated during the group discussions which indicated a need for more training time to learn a tool's usefulness.

DMAAC

TOOL	LIFE-CYCLE	TRAINING LABOR HOURS	TRAINING COMP. HOURS	TRAINING TOOL RUNS	TRAINING USAGE ERRORS	T-BED DEVELOPMENT LABOR HOURS	T-BED DEVELOPMENT COMP. HOURS	T-BED DEVELOPMENT TOOL RUNS	T-BED DEVELOPMENT USAGE ERRORS	T-BED DEVELOPMENT T-BED ERRORS
FAME	REQUIREMENTS	45	4	7	7	79	19	9	16	17
SDDL	DESIGN	35	8	12	7	100	32	27	7	19
TX	CODING	15	5	3	9	128	65	111	45	18
NODAL	TESTING	4	0	0	8	17	3	0	0	0
MAPPER	DATA BASE	8	2	5	0	5	6	12	4	0
OPTIMA	MANAGEMENT	0	0	0	0	0	0	0	0	0
FORMAT	DOCUMENTATION	8	0	0	0	0	0	0	0	0

DMAHTC

TOOL	LIFE-CYCLE	TRAINING LABOR HOURS	TRAINING COMP. HOURS	TRAINING TOOL RUNS	TRAINING USAGE ERRORS	T-BED DEVELOPMENT LABOR HOURS	T-BED DEVELOPMENT COMP. HOURS	T-BED DEVELOPMENT TOOL RUNS	T-BED DEVELOPMENT USAGE ERRORS	T-BED DEVELOPMENT T-BED ERRORS
FAME	REQUIREMENTS	118	36	45	98	79	25	41	77	36
FAME	DESIGN	57	21	22	52	67	21	36	65	30
IS/1	CODING	90	27	21	47	177	68	65	58	54
NODAL	TESTING	0	0	0	0	0	0	0	0	0
IS/1	DOC/MAINT	0	0	0	0	0	0	0	0	0
MAPPER	DOC/MAINT	22	8	4	10	7	5	4	10	0
OPTIMA	MANAGEMENT	0	0	0	0	0	0	0	0	0

Figure 12.1 Activities Summary by Center

DMAAC + DMAHTC - by tool

	TRAINING				DEVELOPMENT				
	L-HRS	C-HRS	T-RUNS	U-ERRORS	L-HRS	C-HRS	T-RUNS	U-ERRORS	T-ERRORS
FAME	220	61	74	157	225	65	96	159	83
SDDL	35	8	12	7	100	32	27	7	19
IS/1	90	27	21	47	177	68	65	58	54
TX	15	5	3	9	128	65	111	45	18
NODAL	4	0	0	8	17	3	0	0	0
MAPPER	30	10	9	10	12	11	16	14	0
OPTIMA	0	0	0	0	0	0	0	0	0
FORMAT	8	0	0	0	0	0	0	0	0

DMAAC + DMAHTC

	LABOR HOURS	COMP. HOURS	TOOL RUNS	USAGE ERRORS	T-BED ERRORS
FAME	445	126	170	316	83
SDDL	135	40	39	14	19
IS/1	267	95	86	105	54
TX	143	70	114	54	18
NODAL	21	3	0	8	0
MAPPER	42	21	25	24	0
OPTIMA	0	0	0	0	0
FORMAT	8	0	0	0	0
	1061	355	434	521	174

Figure 12.2 Activities Summary Totaled

Life-cycle Covered by Questionnaire	Answers Yes/No			Comments Positive/Negative		
	AC	HTC	TOTAL	AC	HTC	TOTAL
Requirements Questions 2-8, 10-12	52/50	69/24	121/74	7/11	10/16	17/27
Design Questions 2-10, 12-14	61/56	84/28	117/84	7/10	4/7	11/17
Coding Questions 2-10, 12-14	42/46	70/35	112/81	6/7	9/10	15/17
Testing Questions 2-8, 10-12	6/41	6/0	12/41	1/4	2/0	3/4

Figure 12.3 Evaluation Questionnaire Response Trends

13.0 TOOL EVALUATION ANALYSIS

This section details the specific characteristics that have been identified as important to a tool's applicability at DMA. In general a tool should be mature, system test should not be conducted by DMA; it should support multiple levels of users, diagnostics and access should be flexible in use with respect to exposure time; and it should be applicable to current DMA practices, so as to not introduce unnecessary work or generate unneeded data. Additionally a tool should support ANSI Standard FORTRAN and/or COBOL; be utilizable in a UNIVAC mainframe or large minicomputer environment; and be wholly supportable within one DMA center's environment.

Further analysis of the data from the tool evaluations provided valuable information for the near-term and far-term modern programming environment system recommendations. The data was first used by GD/DSD in developing evaluation criteria for scoring different implementations which satisfy the same System Operational Concept as defined in the SON/SOC, CDRL A002. These Concept Implementation Evaluation (CIE) sheets are included as Appendix J. The evaluation criteria include the following items:

- Interactive Capability
- Support Documentation
- Diagnostics
  - Documentation
  - Interactive Support
- Automated Procedure
- Maturity
- Vendor Support
- Availability
- Hardware Compatibility
- Environment Compatibility
- Government Access
- Flexibility of Use
  - Hardware
  - Software
- Conceptual Simplicity
  - Tool Use
  - Training
- Output
  - DMA Applicable
  - Understandable
- System Resources
  - Capabilities Supported
  - Allocations Required

Each criterion was evaluated by GD/DSD on a scale from 0 to 10. The meaning of the numeric rating for each particular criterion was defined using terminology appropriate for that topic. These numeric ratings were only one part of a statistical methodology used to generate tool ratings.

Input was sought from DMA and RADC for establishing weighting factors for the evaluation criteria with respect to perceived importance for DMA on a scale of 1 to 3; 3 being of highest importance. In addition, GD/DSD, RADC, DMAHQ, DMAAC and DMAHTC personnel supplied inputs to help define a priority weighting factor for each need from the SON. These are on a scale of 5-1 with a 5 indicating the greatest need and 1 the least. Using the first weighting factor is analogous to performing a coarse rating with the second weighting acting as a fine tuning factor. The particular score for each criterion is the product of the numeric evaluation and the weighting factor. The total score for an implementation is the product of the sum of the individual scores and the need's priority weighting factor. The benefit of this procedure is that it establishes a uniform method of evaluation for all implementations. The total score for an implementation may be low because the need satisfied was not a high priority, very little information about the system was available, or the ratings assigned were low values. The implementations achieving the highest scores formed the nucleus of the tools considered for the near-term modern programming environment. The following definitions apply to the criteria:

EVALUATION RANGES		RELATIVE WEIGHTING FACTORS
High-(H)	= 10-8	3 = Very Important
Medium-(M)	= 7-4	2 = Moderately Important
Low-(L)	= 3-1	1 = Not Very Important
No Information Available	= 0	

13.1 CRITERIA DEFINITIONS

1. Interactive Capability: weight = 3
   - H - Highly interactive
   - M - Partially interactive
   - L - Little or no interactive capability

The interactive capability of a system is a measurement of the amount of manual or batch activity that must be performed when using a system versus how much of the activity may be accomplished through a remote terminal.

2. Support Documentation <u>weight = 2</u>

H - Very thorough and understandable documentation
M - Sufficient documentation
L - Documentation very sketchy (if exists) or hard to interpret

Support documentation must be well organized, easy to interpret and thorough for a system to be used effectively.

3. Diagnostics - Documentation <u>weight = 3</u>

H - Very thorough and understandable documentation
M - Sufficient documentation
L - Documentation very sketchy (if exists) or hard to interpret

Diagnostic documentation is very important to the utility of a system. The documentation must not be cryptic nor require extensive searching when accessed. Additionally, it must be thorough and not be subject to interpretation.

4. Diagnostics - Interactive Support <u>weight = 2</u>

H - Plenty of help available while interacting with tool
M - Sufficient help available while interacting with tool
L - Little or no help available while interacting with tool

Interactive diagnostics must support multiple levels of users including both the novice and the experienced users. Use of the diagnostics should cause minimal interference with the work being processed. The material content must have the same characteristics as the diagnostic documentation.

5. Automated Procedure <u>weight = 2</u>

H - Highly automated
M - Partially automated
L - Not automated

A procedure is defined as a set of activities to be performed in the accomplishment of a task. Automation is a measure of the interaction required by the user of a procedure to initiate the independent activities.

6. Maturity <u>weight = 3</u>

H - Established - well tested through actual commercial use
M - On the market - some commercial use
L - State-of-the-art or newly developed (untested) yet unmarketed

Maturity is a measure of the time a system has been available, the aggregate utilization a system has received, and the state-of-the-art implementated by the system.

7. Vendor Support <u>weight = 1</u>
   - H - Excellent - easily obtained - high quality
   - M - Sufficient
   - L - Poor quality - hard to get

Vendor support is a subjective measure of the capability of a vendor to provide assistance, personal and material, considering items such as physical location, workload, and past experience.

8. Availability <u>weight = 3</u>
   - H - Easily and quickly obtained
   - M - Available - may take a little time to get
   - L - Not available

Availablity refers to the chronological time required for the acquisition and installation of a system.

9. Hardware Compatible <u>weight = 3</u>
   - H - Currently hosted on DMA software development hardware
   - M - Written in a portable language
   - L - Extensive rehost effort required

UNIVAC 1100/62 computers have been identified as the current software development hardware. Tools available on this system may already exist on DMA equipment and/or would not require a rehost effort.

10. Environment Compatible <u>weight = 3</u>
   - H - Applicable to current DMA environment
   - M - Applicable with modification in use
   - L - Not applicable to current software

The current DMA environment may be described by the hardware and software in use and the methodology of their application. Each item must be considered when evaluating environmental compatibility.

11. Government Access <u>weight = 1</u>
   - H - Public Domain
   - M - With restricted rights
   - L - No Rights Available

A public domain system was developed and delivered under government contract and would be available at nominal cost to

other government agencies. Some systems were commercailly developed and may be purchased by the distribution and use of object and source code in a limited rights contract. Other commercial systems only allow the purchase of object code.

12. Flexibility of Use - Hardware weight = 2

H - Portable with respect to DMA hardware
M - Portable in general
L - Not very portable on DMA equipment

A software system may be implemented on multiple computer systems belonging to DMA; exist on one system and be considered highly portable to other hosts; or not exist on DMA equipment and possibly require a rewrite to rehost.

13. Flexibility of Use - Software weight = 2

H - Applicable to most DMA software
M - Applicable to some DMA software
L - Applicable to minimal DMA software

DMA software is comprised of FORTRAN (66 and 77 standards and extensions), COBOL, and multiple assembly languages. Some software systems have applicability across multiple languages or language dialects, but in general will be useful with a specific language implementation. In relative standings FORTRAN is most utilizied, then COBOL.

14. Conceptual Simplicity - Tool Use weight = 2

H - Easily understood/used
M - Understandable/usable with effort
L - Complex in understanding/usability

Tool use simplicity applies only to software tools. A primary consideration is user friendliness. This includes ease of use and understanding as well as interactive support for users with multiple levels of experience.

15. Conceptual Simplicity - Training weight = 3

H - Easily taught/learned
M - Teaching/learning requires concerted effort
L - Complex to teach/learn application

Training simplicity must be considered with respect to DMA on-site capabilities, the background of the personnel involved including education and experience, and the time to be involved in the instruction effort.

16. Output - DMA Applicable <u>weight = 3</u>

H - Complies with current requirements
M - Modifiable to current practices with some effort
L - Not compatible with DMA requirements

DMA has developed formal and informal procedures and standards concerning support of the software life cycle. These procedures have been developed over a long time span and each new system must be evaluated with respect to the impact it would have.

17. Output - Understandable <u>weight = 2</u>

H - Output self-explanatory/summary information supplied
M - Some explanation of output initially required/no summary
L - Extensive training required

The output of the systems must be evaluated for clarity and usefullness. Summary information provided is a consideration. The interpretability of the output, absolutely and relatively, is also a factor in clarity as well as usefullness. Any training required to understand the output and its implications must additionally be evaluated as part of a systems criteria.

18. System Resources - Capabilities Supported <u>weight = 3</u>

H - Supports large number of user/hardware/software interfaces
M - Limited interface capabilities
L - Minimal interface capability

The resources supported by a system, terminals, users, tape/disk drives, specialized peripheral devices, etc. are important to DMA due to the large number of users and the multiple architectural devices/interfaces utilized.

19. System Resources - Allocations Required <u>weight = 3</u>

H - Minimal memory/cycles/special equipment required
M - Limited impact on resources
L - Heavy resource utilization

When evaluating a system which will be used concurrently with or integrated into other system software, computer resource allocation should be a major consideraton. Specific areas of interest include combinations of memory, cpu cycles and specialized equipment required.

14.0 ALTERNATIVE ANALYSIS

The Alternative Analysis was the next step in the contractual effort. The results of the Tool Survey and the SON/SOC are combined and analyzed to formulate the Best-Case model for the DMA modern programming environment (MPE). This Best-Case MPE (reference Section 15.0) is an unconstrained model; it provides the baseline for the formulation of the Near-Term and Far-Term MPE's. Constraints such as schedule availability, cost, hardware compatibility, user-friendliness, technical capability, etc. are applied to the Best-Case MPE to arrive at the Near-Term MPE. The detailed discussion of this analysis is contained in Section 16.0. As constraints are relaxed, and new technologies become available, the Far-Term MPE is defined. Section 17.0 describes this Far-Term configuration. In Section 18.0 candidates for future research and development activities are recommended to fulfill DMA needs that are not satisfied by available technology and tools. Finally, cost data for the Near-Term and Far-Term MPE's is contained in Section 20.0.

15.0 BEST-CASE MPE

Figure 15.1 presents the evolutionary process through which the Best-Case MPE and Near-Term and Far-Term Modern Programming Environments were developed. The first step in the process involved developing Concept Implementation Evaluation (CIE) sheets and a CIE matrix. This matrix was developed to provide continuity with the SON/SOC document. It contains the same data as the SON/SOC matrix with the addition of an "I" where one or more implementations of a specific need-concept relationship exists. Other implementations may exist but were not discovered during the course of the project. An "X" indicates the absence of a known implementation. The CIE matrix is presented in Appendix G. As described in Section 3.0, the identified needs of DMA were assigned weighting factors. During the Tool Survey task of the contract, evaluation criteria for system operational concept implementations were established and assigned weighting factors with respect to the importance of the criteria within the DMA environment. Implementations of specific concepts were then numerically rated against the criteria by GD/DSD. The relationship of multiplying the criteria weight times the criteria rating, summing the ratings, and multiplying the total by the need weight, which derived an implementation score, was formatted into a CIE sheet. This process, the criteria, and the weighting factors are explained in Section 13 and the sheets for all identified implementations are attached as Appendix J.

```
**********                **********                *************
*NEED &  *                *TOOL    *                *DSS        *
*CRITERIA*                *SURVEY  *                *TOOL       *
*WEIGHTS *                *CRITERIA*                *EVALUATIONS*
**********                **********                *************
    |                          |                          |
    |                          |                          |
    |                     **********                      |
    |                     *CIE     *                      |
    --------------------->*MATRIX &*<----------------------
                          *SHEETS  *
                          **********
                               |
    <------------------------------------------------------>
    |                 |                  |                 |
*********        *********          *********          *********
*TOOL   *        *NEED   *          *CONCEPT*          *SCORE  *
*RANKING*        *RANKING*          *RANKING*          *RANKING*
*********        *********          *********          *********
    |                 |                  |                 |
    |                 |                  |                 |
    |             ******            *********              |
    |             *NEED*            *CONCEPT*              |
    |             *BEST*            *BEST   *              |
    |             *CASE*            *CASE   *              |
    |             ******            *********              |
    |                 |                  |                 |
    -------------------------->-<---------------------------
                               |
                            ******
                            *BEST*
                            *CASE*
                            ******
                               |
                               |
                         ***********
                         *NEAR-TERM*
                         ***********
                               |
                          **********
                          *FAR-TERM*
                          **********
```

Figure 15.1 NEAR-TERM/FAR-TERM ENVIRONMENT EVOLUTION

15.1 GENERAL METHODOLOGY

Utilizing the scores of the CIE sheets as a basis for comparison of implementations, four rankings were developed: by tool, by need, by concept, by score. These rankings are presented in Appendix H.

The ranking by tool provided insight into each tool's ability to resolve multiple needs as well as the importance of the needs satisfied. By ranking the needs the best implementation to solve each need was easily identified. This was also the reason for generating the concept ranking, to identify the best implementation of each concept. Finally, rankings by high score and specific tool were established. A direct relationship exists between an implementation's score and its productivity within the DMA environment; however this is only one of many considerations.

The best (highest scoring) implementations for each need and each concept were used to generate Best-Case-by-Need and Best-Case-by-Concept environments respectively. These two environments were very similar. A generalized Best-Case MPE was generated by combining the need and concept environments and using information available in the tool and score rankings. All three environments are included in Appendix H.

The Best-Case MPE was then evaluated in multiple steps for near-term considerations. In step one the Ada language, for structured programming, the Ada Programming Support Environment (APSE), as an integrated support development system, and the User Interface for On-Line Assistance (UIFOLA) were all deleted as being for far-term consideration only. SOFTOOL 80 was deleted as having minimal capability when compared to cost and need to be satisfied, as was CPAT and the PlanIt Billback system. PRICE-S was deleted due to a time-share only availability. FAVS (or its commercial equivalent RXVP80) and FORTRAN 77 were included in the near-term due to their current application and their high scores. The MPE administrator and toolsmith functions were included on the basis of the ease of implementation, and the potential benefits to the implementation of the MPE as a point source of information to the user community. HYPERGRAPHICS was chosen for its flexibility in use as a training tool, and its relatively inexpensive cost.

15.2 TOOL SELECTION EXAMPLE

In Appendix G, the CIE Matrix, there is an 'I' located at the intersection of concept #13 and need #57, under "REQUIREMENTS". This identifies that a need was expressed in

the SON/SOC document for a software development tool to be used in automating the requirements generation process. Further, the 'I' specifically implies that one or more implementations of tools to meet this need are presented in the CIE sheets, Appendix J. The CIE sheets, of which there are 173, provide specific information about a tools score with respect to the criteria listed previously in this section. The sheet also serves as a tabulation form for generating a tool's total score for an implementation as previously described by incorporating coarse and fine weighting factors. The sheets are primarily collated by SOC number, then by need number within each SOC. Multiple implementations within each SOC/Need sequence have not been collated.

Turning to SOC 13, Need 57 in the CIE sheets it will be discovered that five implementations have been identified: FAME, RDP 1100, PSL/PSA, SRIMP, LARE; with respective scores of 1289.4, 722.4, 1306.2, 961.8 and 1247.4. RDP 1100 and SRIMP were eliminated due to their low scores relative to the other tools. By determining the best tool (highest score) to satisfy each need, PSL/PSA is selected. A similar evaluation for the best tool to satisfy each concept will have the same result, which eliminates FAME and LARE as possibilities. Now PSL/PSA must be evaluated within the constraints of the Best-case MPE.

USE.IT was identified as an implementation of the SOC 11/Need 57 relationship. FAME is a subset of USE.IT, hence PSL/PSA and USE.IT have duplicating features. Inspection of the CIE BY SCORE listing shows values of 3172.2 for PSL/PSA, 3131.4 for FAME and 2350.0 for USE.IT. This would imply PSL/PSA would be the chosen tool, except for the fact that USE.IT has all the functional capabilities of FAME, so USE.IT has a functional value of 5481.4. Additional information was sought at this point to verify the selection of USE.IT.

DMA already had PSL/PSA, although it was not being widely utilized, hosted on UNIVAC equipment. However, the information obtained from the ISDOS project at the University of Michigan indicated the UNIVAC version was not well supported; and that future benefits will be introduced via the VAX architecture. USE.IT was already hosted on a VAX, as was PSL/PSA. Further investigation provided the deciding factor. USE.IT had an interactive user-friendly front end with graphics capabilities. A PSL/PSA front end equivalent to the USE.IT system is under development by the ISDOS project. However, it is a student based development and may not be available or supported for an indefinite period. USE.IT also has the capability of producing FORTRAN, PASCAL or, in the near future, COBOL code. PSL/PSA is being ex-

tended to produce FORTRAN code; but again it is a student based effort. Additionally, the Army is utilizing USE.IT in its Ada development efforts. Through this process USE.IT was selected to support the SOC 13/Need 57 automated requirements generation need.

15.3 TOOL BEARING HOST SELECTION

The second step involved identifying a tool bearing host (TBH). A large selection of mini and mainframe computers including most major brands hosting FORTRAN, COBOL and supporting software development tools were considered; however, only Harris, SEL, UNIVAC and VAX equipment were found to provide the type and extent of support required by DMA. A major requirement of the MPE concept was the availability of programming support environment tools. Other considerations were vendor support, DoD R&D efforts, physical size, architectural capabilities, and cost. The phase II evaluation effort described in Section 10.0 determined the only viable options were VAX and UNIVAC. The Harris system's main drawback was a 24 bit word size which would require a repacking mechanism when converting code for use on 16 bit production units or the 36 bit UNIVAC systems. Additionally, most tools on this system supporting life cycle phases other than programming were not mature. The SEL hardware's 32 bit word size and through-put capabilities were impressive, but the same problem existed with support software. Performing an analysis of the minimum system configurations available in the near-term concluded with a recommendation of a VAX as the near-term tool bearing host.

In addition to the statistical data available in the Appendices, the major considerations were the quantity and quality of software development tools available on the systems, and the desire to remain state-of-the-art while evolving a far-term environment. Though UNIVAC was strongly considered as a TBH for the MPE due to its current use within DMA as a production and development machine, the quantity of software development tools and the state-of-the-art capabilities of tools currently hosted on the VAX are not available on UNIVAC. As examples, Ada and its support environment is currently being hosted on a VAX. There is no known activity proposed for rehosting APSE to a UNIVAC. In the Software Configuration Management seminars being conducted by the Data Processing Management Association (DPMA) many of the manual techniques recommended have already been automated by the Programmer's Work Bench (PWB) being marketed under VAX/VMS control. In the area of requirements specification the four major systems that are available or are under development: PSL/PSA, MEDS, RSL/REVS and USE.IT all

are hosted on VAX systems. Only PSL/PSA is currently on UNIVAC equipment; but according to the ISDOS project at the University of Michigan it is not as well supported as the VAX implementation nor does it have as many capabilities. The NBS recently released the FORTRAN 77 Analyzer. NBS developed its own user interface on a VAX. As specifically relates to the recommended MPE, the capabilities provided through the use of USE.IT and the IS/1 PWB are not currently available on Univac equipment.

Also, information was gathered by meetings with DMA personnel, during recent contract extension activities, that indicates a move within both centers toward the use of VAX machines. Within the next year DMAHTC is to obtain at least 8 and DMAAC 7 VAX systems. These will be VAX-11/780's for the most part and many will be delivered with products. For examples, DMA is to obtain the following VAX system based products: CPS Clustered Carto, TES/EMPS, Terrain Edit/Elevation and possibly the CPS Clustered Carto system. DMA will eventually have to provide maintenance for these systems.

Additionally, there are benefits to be derived through the use of a separate machine dedicated to software development. The importance of production runs normally results in a secondary priority for development runs thus reducing their chances for a faster turnaround. Moving development work off the production machine(s) results in (1) better response time for checkout and development runs and (2) less interference with production work.

Aside from life cycle system development support tools the following performance statistics were considered. For 'throughput' comparison, an article in Datamation, November 1980, indicated the KOPS (thousands of operations per second) rating of the VAX-11/780 to be higher than an IBM 370/158, a DEC 2050, or UNIVAC 1108 or 1100/60 C2 computers. The Whetsone benchmark comparisons in KIPS (thousands of instructions per second) at single and double precision operations support this data. The benchmark indicated the VAX-11/780 outperformed the SEL 32, IBM 370/155, DEC 2050 and various V77 computers in double precision mode, while only being outperformed by the DEC 2050 in single precision mode. Figure 15.2 illustrates the recommended near-term MPE utilizing a VAX configuration.

The majority of recommended tools were developed on the VAX-11/780. These tools are upward compatible although downward compatibility is not assured.

The recommended tool set addresses the significant needs of DMA as identified in the SON/SOC document. Ten of the thirteen most important needs, as determined by DMA, RADC and GD/DSD, have implementations identified in the Near-Term MPE which will immediately enhance DMA's capabilities.

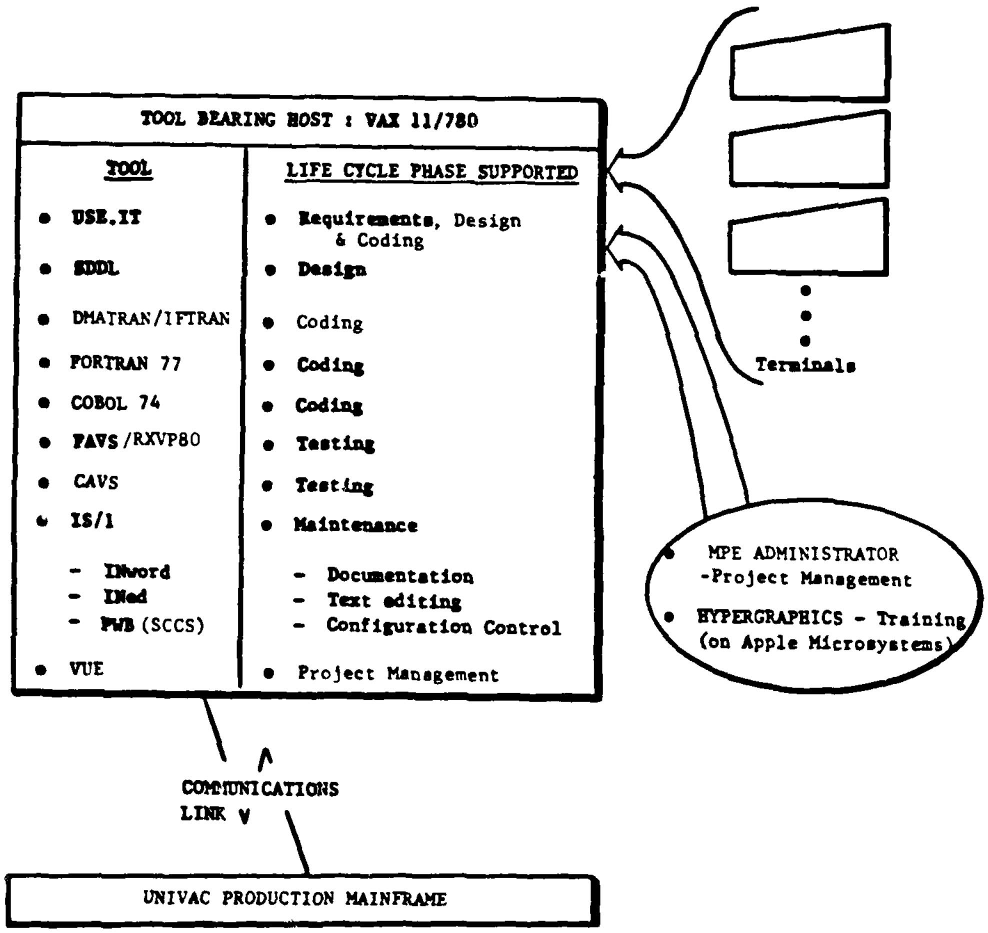


Figure 15.2 Near-Term System Configuration for DMA Modern Programming Environment

16.0 NEAR-TERM DESCRIPTION

The near-term system was selected through the previously described process to meet the immediate needs of DMA. As defined the system has a high probability of improved productivity. A more detailed description of the Near-Term MPE is provided in the Functional Description and System/Subsystem Specification annexes to this report. The specific VAX system configuration recommended evolved through information obtained internally within General Dynamics, data provided by DMA, consultation with DEC representatives, and consultation with a major user of VAX systems. For clarification, 'maintenance functions' is defined as post production software development activity requiring work in one or more phases of the life cycle: requirements, design, programming, testing.

16.1 DEVELOPMENT ENVIRONMENT

A VAX-11/780 will be utilized for the entire software development life cycle including requirements, design, programming, and testing, as well as configuration control and project management activities. The specific configuration is described in the System/Subsystem Specification, CDRL A007.

All software development is performed under the control of the project management tool, VUE. Upon receiving a job request, the project management tool is initiated for the job and at various points in the scenarios, the project management system is updated to reflect pertinent decisions and actions. Examples of the inputs and outputs for the VUE system are illustrated in Figure 16.1.

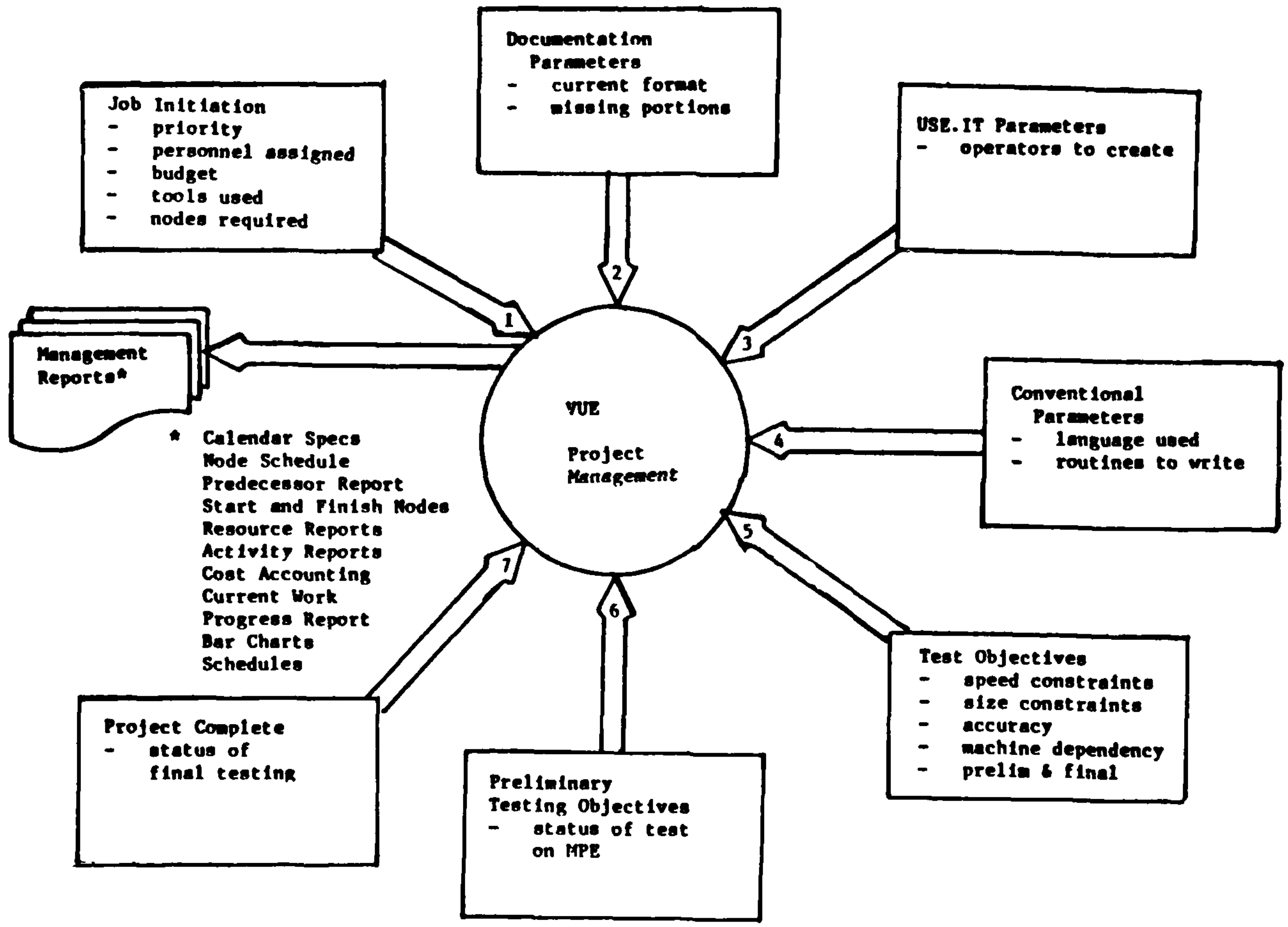


Figure 16.1 MPE Project Management Overview

For purposes of discussion, scenarios will be considered for the following categories of software development:

1) maintenance of existing software which has not been upgraded by the Software Improvement program (SIP)

2) maintenance of existing software which has been SIP upgraded,

3) software presently under development for which standards were not specified,

4) new software to be developed by DMA for which standards will be specified, and

5) new software to be developed by contractors for which standards will be specified.

The techniques discussed are intended to demonstrate the applicability of the recommended tools to the various scenarios. Specific usage methodologies will be developed during the MPE system implementation as outlined in Section 19.1.

The application of MPE tools to the DMA environment is illustrated in Figures 16.2 and 16.3.

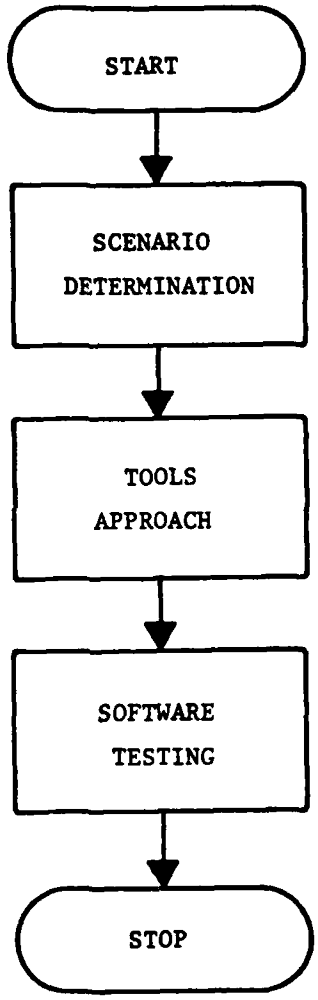


Figure 16.2   MPE Scenario Overview

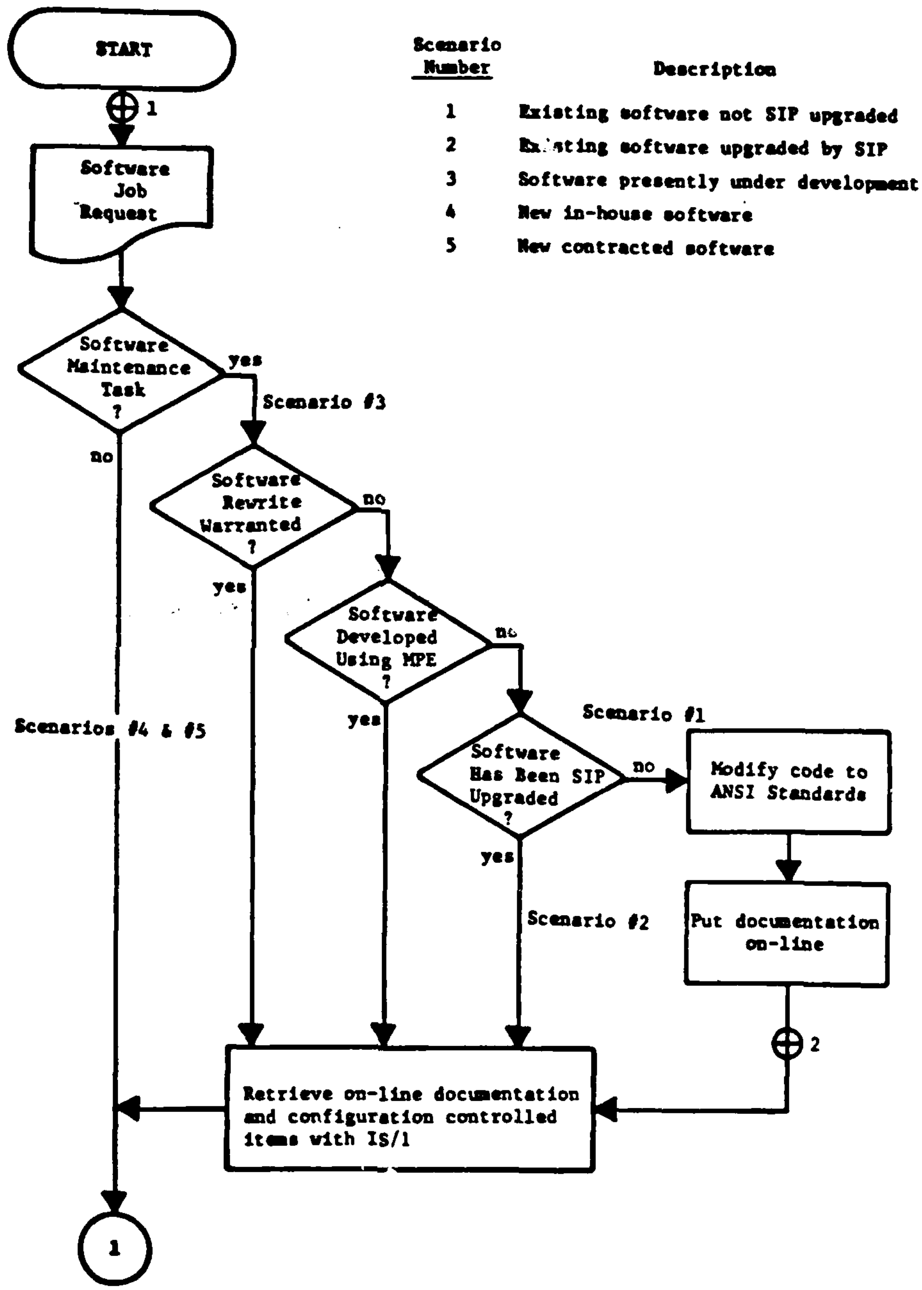


Figure 16.3 MPF Scenarios
(page 1 of 2)

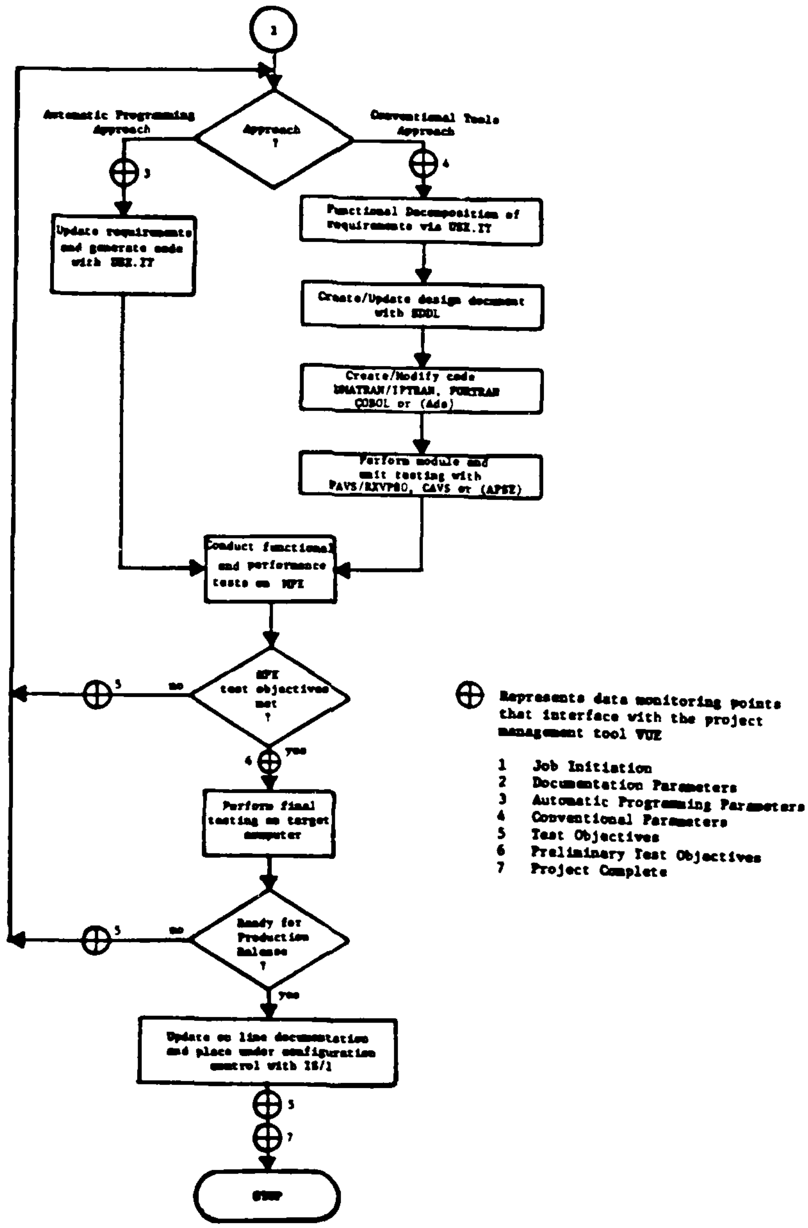


Figure 16.3 MPE Scenarios
(page 2 of 2)

16.1.1 Approaches

Within the defined scenarios, one of two basic tool approaches will be utilized. These are described in the following two sections.

16.1.1.1 Automatic Programming Approach

The first, referred to as the "automatic programming approach", will make repeated use of the subsets of the tool USE.IT until performance criteria are achieved. The usage of the various subsets is as follows:

- the USE.IT graphics editor is used to enter program structures, called control maps, to functionally decompose requirements and design specifications as well as changes, if any, which are required as a result of performance testing,
- the Analyzer verifies internal consistency and interfaces,
- the RAT automatically produces programs from Analyzer output,
- source produced by the RAT is compiled and linked, and
- the system is performance tested to determine acceptability.

Failure to pass performance testing results in repetition of these steps until criteria are satisfied.

There appears to be no practical limit to the size of system which may be developed with USE.IT. As systems are developed via USE.IT, generic operations are developed and can be placed in a library for use as building blocks on subsequent systems. For this reason, detailed documentation within AXES statements is considered mandatory.

16.1.1.2 Conventional Tools Approach

The second, referred to as the "conventional tools approach", will make use of the SDDL, DMATRAN/IFTRAN/FORTRAN or COBOL, and FAVS/RXVP80 or CAVS tools through the life cycle. Utilization of tools in the "conventional tools approach" consists of repeated application of the following procedures until performance criteria are achieved.

- USE.IT is utilized to functionally decompose the requirements specifications and to check the data flow and interfaces on the resulting program model.

- SDDL is used to originate the design or make design changes, if any, which were mandated as a result of performance testing

- Source code (DMATRAN/IFTRAN/FORTRAN or COBOL) is modified to reflect changes brought about by design changes, performance changes, or FAVS/RXVP80 or CAVS evaluation

- FAVS/RXVP80 or CAVS are envoked for the purpose of detecting syntax errors, performing static analysis, and performing execution analysis

- Performance testing is evaluated to establish the acceptability of the system. Failure to pass performance results in repeating the process.

16.1.2 Scenarios

One of these tool application approaches is followed until the preliminary test objectives are met. At this time, the source is transmitted via data link to the target host for final testing.

While testing on the target host, the project management system is apprised of the test status. Upon successful completion of final test objectives, job completion data is processed by the project management system. This action prevents the system status from being obscured from control and insures a match between production software and the associated documentation. Target host test objectives will verify machine dependent devices and techniques. Once final testing is completed and the system is ready for production status, on-line documentation such as requirement and design documents, source code and test data should be updated and placed under configuration control using SCCS. Under the conventional approach all coding will be accomplished in ANSI X3.9-1978 FORTRAN (77) or ANSI X3.23-1974 COBOL (74). The code should be structured using the SEQUENCE, DOUNTIL, DOWHILE, CASE control constructs. For FORTRAN programs the DMATRAN precompiler would be used to translate the structured code into ANSI standard code prior to final compilation and test on the target production machine. Under the automatic programming approach ANSI standard code is produced.

16.1.2.1 Existing Software

Upon receiving a maintenance job request, the project management system is provided with sufficient information to make an entry for the job. Should the job request include significant requirements modifications or is for a system which was developed since the MPE installation, "the USE.IT method", as described above will be utilized to rewrite the system.

For job requests requiring no major requirements changes and representing systems developed prior to the MPE, the SIP upgrade status of the system is determined and action taken as described in the following scenarios.

16.1.2.1.1 Not SIP Upgraded

Job requests for systems which have not been enhanced by the SIP program are analyzed for the level of effort required to bring them to SIP standard. This level of effort is compared to that required to express the system requirements and generate the system by application of USE.IT. If the effort required to bring the system to SIP standards matches or exceeds the effort required to re-write with USE.IT, the system will be redeveloped by the USE.IT method as described above. Otherwise, project management entries will be made to reflect the SIP upgrade effort and the system will be brought to SIP standards with tools and methods of the SIP program. Once the SIP upgrade has been accomplished, the system will then be updated by use of MPE tools and methods as described in the description of the "conventional tools approach".

16.1.2.1.2 SIP Upgraded Software

The SIP program is intended to consolidate into a single coordinated program many on-going, related, DMA activities. One of these activities is the improvement of existing UNIVAC software. Job requests pertaining to systems having been upgraded by the SIP program proceed through the previously defined conventional development process. The SIP program is intended to consolidate into a single program many ongoing, related DMA activities including an effort to improve existing UNIVAC software. The check-out process is initiated by updating the project management system to reflect usage of SDDL, FORTRAN or COBOL, FAVS/RXVP80 or CAVS, and IS/1 before proceeding to the check out process. Upon achieving test objectives, on-line accumulation of documentation including requirements, design documents, source code, and test data is accomplished. The accumulated documentation is then placed under configuration control, using SCCS, the project

management system is notified of the completion of the MPE test objectives, and the job is transmitted to the target host for final testing as described above.

16.1.2.2 Software Under Development

As a result of the SIP program, systems under development during the transition to MPE will be required to conform to programming and documentation standards. At implementation time, software contractors will not necessarily have access to the MPE tools. For this reason, it is recommended that software under development by contractors during the transition to MPE be developed as contracted. In anticipation of this action all contracts should include a standard documentation definition and a requirement to furnish documentation on a media readable by the MPE. Such documentation will be placed under configuration control.

Those systems under development by DMA during MPE implementation, should be classified by priority, size, complexity and level of expended effort. High priority systems and those on which a high percentage of estimated effort has been expended should proceed as originally planned. Care should be taken to insure that these systems conform to standards with the final documentation being placed under configuration control. The remaining systems should be processed by the "automatic programming approach", as described above. The development of these systems will provide invaluable data in the determination of the cost effectivity of USE.IT. Should there be any systems for which USE.IT development appears impractical, they should be developed with the "conventional tools approach" of the MPE, as described above. Action taken in this development process is described in the scenario for SIP Upgraded Software.

16.1.2.3 Future System Development

To establish uniform systems which will be more readily maintained through their life cycle, every effort should be made to have systems produced that meet rigidly enforced standards. It is understood that these standards are currently under development and specific MPE related standards will be incorporated as required during final MPE implementation efforts.

Since systems operated by DMA are developed internally to DMA and externally by software contractors, the development of future systems is discussed in the following two scenarios, Systems to be Developed by DMA, and Systems to be Developed by Contractors.

16.1.2.3.1 Systems Developed by DMA

Systems developed internally to DMA should make application of USE.IT. This utilization causes the accumulation of a library of operations as well as putting development personnel in an environment in which structured system development is enforced.

The application of USE.IT proceeds as described above. Successful completion of preliminary test objectives results in reporting of this status to the project management and the on-line accumulation of documentation as described in the scenario for SIP Upgraded Software. If, for any reason, the application of USE.IT is not cost-effective the development will proceed as described in the scenario for SIP Upgraded Software and final testing on the targetted host is performed prior to completion of the job request.

16.1.2.3.2 Systems to be Developed by Contractors

Systems which are developed by contractors should be done in the same manner as those developed internally. As a minimum, contractors should be required to adhere to the programming and documentation standards established for DMA with all documentation placed on a media readable by the MPE. Ideally, contractors will have access to the MPE tools and will be required to follow the methodologies established for DMA.

16.2 SUPPORT ENVIRONMENT

The MPE adminstrator and toolsmith functions will support the project management function as well as system management and training; and HYPERGRAPHICS, a tool for building presentations and interactive lesson plans will be utilized for training purposes. The selection of HYPERGRAPHICS is based on its simplicity in use as well as cost and physical availability. The generation of visual material to support a training document is easily performed and maintained. A training program for Ada/APSE, a possible part of the far-term environment, has already been developed on this system. The system was developed in a university environment and is not expensive; and the capability of hosting on a microcomputer allows physical access in almost any area.

The VAX computers and UNIVAC production mainframe will need to be connected through a communications link using a standard protocol or over an I/O channel. To support users in a timely manner and to provide adequate access, multiple VAX computers will be required. The use of multiple systems en-

courages the placement of MPE systems within functionally displaced areas instead of a centralized location thus allowing easier access to the system. A recommendation of three identically configured systems, resident at each center and intra-center connected by DEC communications equipment, is considered sufficient to support current activities. In the future multiple systems could easily be added as required. Experience within GD/DSD was one of the primary information sources used in deriving the recommended number of VAX systems. Using well established software development areas as examples it was determined how many terminals were required to support a given number of programmers. Next, given the number of programmers currently working within each DMA center, the number of interactive terminals required within each center for the near-term was calculated. Finally, using both VAX manuals and discussion with DEC representatives it was determined that 3 systems would optimally support this number along with the software development environment.

16.3 RELATIONSHIP OF MPE TO SIP

Preliminaly findings indicate that there is little conflict between the Software Improvement Program (SIP) and the recommended MPE. The only potential conflict noted thus far is the purchase of a configuration control system under SIP. This could result in a duplication of the capabilities to be provided by SCCS within the IS/1 PWB thus decentralizing such control over production software systems. Other than this the two activities SIP and the MPE seem to complement each other. For example, both the SIP and MPE support the use of FAVS and CAVS for programming development and the use of ANSI standard code. They also each support the use of life cycle phases, structured programming practices and automated tools during software development. The MPE will benefit from the SIP program in several ways. As examples, the development of a set of software development standards by DMA under SIP is supported by the MPE and the preparations being made under SIP for the skills upgrade will benefit the training/introduction efforts of the transition to recommended near and far-term MPE's. The SIP program will also benefit from the establishment of the MPE. Documentation updates required by SIP for production program upgrades can be put under IS/1 for configuration control. A USE.IT support library may be developed as a subset of the centralized library which was established under the SIP program. Such a library may be used to enhance the rapid prototyping capabilities of USE.IT as described in Section 16.4.

16.4 USE.IT TOOL EVALUATION

USE.IT is a tool based upon a methodology of successive decompositions of requirements and design specifications. Information which is known about the problem is entered into the solutions' model. Work needed to complete the model is apparent since the solutions structure is known to be a tree with one root node and leaf nodes being primitives or well defined operations.

Each data type has associated with it a set of non-decomposable primitive operations (primitives) which manipulate the data for this type in some consistent manner. For example,

(a) data type "integer" has primitives ADD, SUB, IMUL,IDIV;
(b) data type "vector" has primitives rotate and vector cross-multiply;
(c) data type "screen" has primitives erase, plot, point, and unplot.

These are already available in the USE.IT library. The user is able to create a new set of data types whose definitions may use previously defined types.

Operations such as

- DRAW BOX which draws a rectangle,
- DRAW HEXAGON which draws a hexagon,
- DRAW HYPERBOLA which draws a set of hyperbolas, and
- MIRROR which given a point in one quadrant will generate the image points in the other three quadrants by reflection about the x axis, y axis and the origin,

are examples of library functions that can be constructed to satisfy the requirements of a particular user problem. Such library functions may be maintained in a support library for utilization on other applications.

The constructs of USE.IT provide highly structured models where data flow is strictly controlled. It is this control that allows the model to be inspected for correctness. At any point in time the model can be analyzed for completeness and correctness.

When the solution has been decomposed from the root node to the leaf primitives or operations, the model is complete.

If the inputs, outputs and structures no longer produce errors in the analysis phase, the model is correct and FORTRAN code can be automatically generated to produce the solution. Evaluation results suggest that any problem can be decomposed employing USE.IT. A LORAN navigational lattice problem was

used for the evaluation model. A complete problem definition can be found in Appendix I.

For the LORAN problem, 23 man days were spent in preparation for formal presentation of the problem solution. Of these, 17 were spent in development of user level operations while 6 were used in applying the operators to the problem solution.

Several findings concerning the USE.IT tool have already been obtained using the LORAN problem as a test-bed. First, the USE.IT tool forces structured development through the use of its constructs, constructs which are best understood by programmers. The USE.IT tool requires the software developer to concentrate on what is to be accomplished instead of having concerns over how the task is to be accomplished. For best use of USE.IT, patterns of thought must match the tool's approach of decomposition. Individuals who already produce structured code should find USE.IT easy to use.

The FORTRAN code produced by USE.IT

(a) is error free,
(b) is not structured, although FORTRAN 77 constructs may be modeled with AXES,
(c) contains very few comments,
(d) has variable names which are nondescriptive,
(e) and is inefficient as compared to code produced by an experienced programmer.

However, the maintenance of USE.IT produced FORTRAN code would defeat the purpose of the tool since respecification is the correct way to maintain a model.

To reduce inefficiencies in the FORTRAN code, higher level operations can be coded and placed in a library. This approach is highly encouraged since a library of models can reduce the detail of specification, reduce repeated production of the same model, and reduce the number of subroutine calls.

17.0 FAR-TERM DESCRIPTION

The recommended software tools and hardware tool bearing host (TBH) for the Far-Term Modern Programming Environment are shown in Figure 17.1. The recommendation of the VAX-11/780 as the far-term tool bearing host is based upon two important facts. First, the VAX-11/780 will already be on-site and available as the TBH because it is also recommended in the near-term. Consequently, the near-term to far-term transition will be smooth and will not disrupt DMA software development and maintenance activities. Secondly, the trends in the software industry are to host many development aids and tools on a VAX system. Examples are the Ada language compiler and the Ada Programming Support Environment (APSE). Therefore, it is anticipated that DMA will have ready access to future tools that will be developed and hosted on the VAX.

17.1 SOFTWARE TOOLS

The recommended software tool suite supports all the software development life cycle phases. The USE.IT, SDDL, DMATRAN/IFTRAN, IS/1 (with INword, INed, and SCCS), FORTRAN 77/COBOL 74, and FAVS/RXVP80 or CAVS tools support the requirements, design, coding, maintenance (documentation, text editing, and configuration control), maintenance coding, and testing tasks, respectively. All of these tools will be in-house and operational on the VAX-11/780 from the near-term environment.

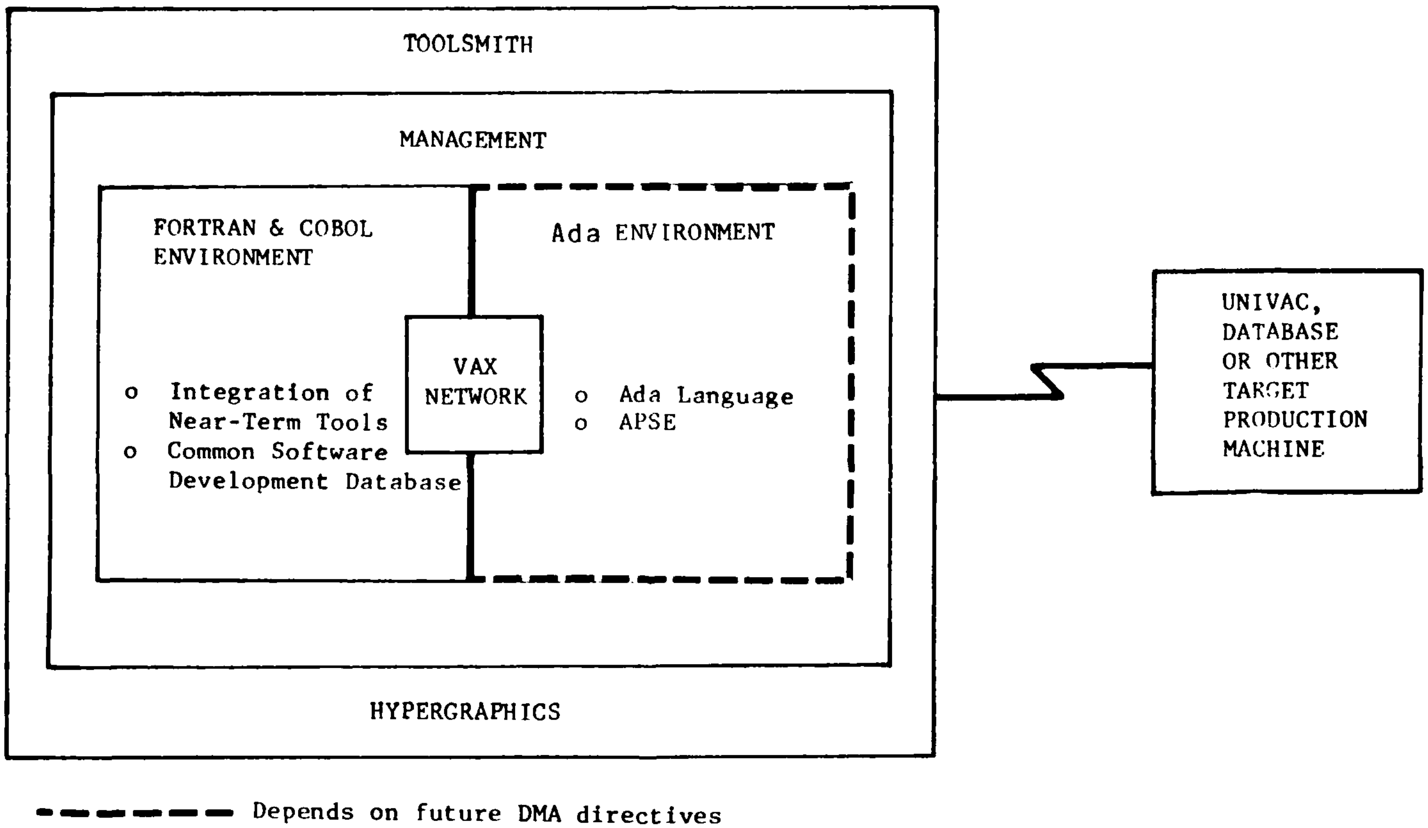


Figure 17.1 Far-Term System Configuration for DMA Modern Programming Environment

17.1.1 Ada

Ada, the DoD standard high order language, is scheduled to have a compiler hosted and targeted to the VAX by the beginning of 1983. The MAPSE (Minimal Ada Programming Support Environment) consists of a loader, data manager, compiler, editor, linker, and command interpreter all integrated into a smoothly working tool set. The APSE consists of the MAPSE plus user specific tools. The architecture of the APSE is designed to permit easy integration of user tools. The MAPSE on a VAX is scheduled for completion by the end of 1984 and should be available for the far-term environment. It is recommended that these Ada capabilities be included in the far-term environment so that DMA can benefit from this new technology and from the Ada tools produced by industries, government, and universities.

17.1.2 Rehost Efforts

In the far-term environment it is recommended that one tool be rehosted from the near-term environment to the VAX. The HYPERGRAPHICS tool would probably be an easily rehosted software package. HYPERGRAPHICS was already recommended in the near-term and would be a proven, mature tool. Also, as mentioned previously, an Ada/APSE training program was developed on this system that could be used as an aid in transitioning to the far-term MPE.

17.1.3 Development Efforts

One effort involved in the upgrade from near-term to far-term MPE will be the development of logical, automated interfaces between the life cycle development tools. This means that the output from one tool should be automatically matched to the input required by the next phase's tool. This could involve tool modifications and/or the addition of postprocessors or preprocessors to be used between phases. Essentially the resulting system would appear to the user as 1 tool with 1 interface. Additionally, a centralized software development database might be developed as a common information source and storage area for use by all life cycle development tools.

Also, at this time a fully satisfactory project management tool for the VAX has not been identified. The particular capabilities of a software cost estimating tool and a chargeback accounting system could be obtained in one of two ways. First, a full capability project management system for the VAX may be developed by a software vendor and it may satisfy DMA needs. If so, it can be acquired and used. The

other option is to contract the development and installation of a project management and accounting system that is particularly tailored to the DMA organization. There is a definite need for automated tools to support project management in the far-term MPE.

Finally, methods should be developed to communicate through the software development MPE systems with commonly used databases already existing at the DMA centers.

17.2 KEY POINTS

In summary, the key points in the Far-Term Modern Programming Environment are:

(1) Use of the VAX-11/780 as the tool bearing host

(2) Carry over of the proven near-term tools

(3) Introduction of Ada to prepare for future technology and tool development if DMA elects to use Ada

(4) Continued support of the FORTRAN and COBOL environments

(5) Logical integration of the life cycle development tools

(6) Common formatted software development database

(7) Ability to write code for additional DMA production machines besides UNIVAC and VAX.

This plan offers to DMA the benefits of low cost for the far-term environment and the minimum risk in technical and schedule areas.

18.0 R&D ACTIVITY

This section describes new software engineering tools to better satisfy DMA needs as well as modifications to existing tools. The purpose of these research activities is to fine-tune the Far-Term MPE to maximize efficiency in the software development process. Anticipated benefits include maximizing the productivity of programmers, eliminating redundancy among projects by establishing libraries of reuseable software, providing a capability to accurately project and track development costs, and being in a position to reap the benefits of other DoD research activities. Other potential benefits include the identification of previously undiscovered needs at DMA, and the capability to accomplish yet undefined software tasks without a major effect on the production environment.

18.1 NEAR-TERM MPE SPECIFIC

The Near-Term MPE is a set of tools with specific uses recommended. These tools have potential utilization beyond the scope of this study. These potentials should be investigated for application to the Far-Term environment.

18.1.1 IS/1 System Implementation

The programmers work bench (PWB) uses two main tools, SCCS and MAKE. Both tools will need default values and implementation parameters specified with respect to the DMA environment standards established. These items invoke specific compilers, enforce naming conventions, update release versions, provide for limited access and numerous other functions which must be standardized across all systems as well as molded to specific DMA requirements. A supporting methodology for utilization of these tools must also be developed.

18.1.2 USE.IT

For reasons of compatibility, the USE.IT developer should be contracted to produce FORTRAN 77 source code. It should be noted that USE.IT produces FORTRAN 66 code which is a proper subset of FORTRAN 77. The constructs available under FORTRAN 77 may be developed under USE.IT for use in developing program/system models utilizing USE.IT.

A USE.IT operation library should be developed and maintained as a subset of the program support library. Criteria for inclusion in the library should be established. Rapid

prototyping within the DMA MPE is dependent on the accumulation of a set of DMA specific operations.

Although USE.IT is the most promising tool available to accomplish the automatic programming approach, further research is required in the areas of training, DMA programmer acceptance, time critical systems performance, commitment of developer to produce other source languages, and the ability of the developer to provide long-range support. Access to USE.IT by potential DMA users for test-bed experimental development efforts would be very beneficial in researching these areas. Emphasis should be placed on the interface between the two high level tools, USE.IT and SDDL. Although preliminary considerations indicate the feasibility of the task, further research is required to implement the interface.

18.1.3 VAX-11/730 Experimental System

To provide convenient user access and the most cost effective implementation of the experimental system, the ability to host the the MPE tools on the VAX-11/730 should be considered. Points to be researched include: (a) ability to host recommended MPE tools on the 11/730; (b) expected performance ; and (c) extent of possible upgrade.

18.1.4 FAVS/RXVP80

DMATRAN and FAVS were created from earlier versions of the General Research Corporation IFTRAN and RXVP80 software tools in an effort to customize them to fit the DMA environment and UNIVAC systems. Since the creation of DMATRAN and FAVS, the commercial versions IFTRAN and RXVP80 have undergone many upgrades and changes. Though these commercial versions are available on the VAX, there is some question as to their applicability to the DMA environment. The acceptability of IFTRAN and RXVP80 to satisfy DMA requirements should be researched before decisions can be made whether to rehost DMATRAN and FAVS or purchase their commercial counterparts.

18.1.5 Advancements in the State-of-the Art

The recommended tools and hardware represent the current state-of-the-art at the time of this writing. Due to their present technological position, it is anticipated that they will remain on the leading edge of technology. Obviously, should this position change prior to implementation of the DMA MPE, systems should be acquired that are deemed to best fit DMA needs at that time.

18.2 FAR-TERM MPE SPECIFIC

The Far-Term MPE is based upon software development tools expected to be available by 1987 and their relationship to the tools and hardware recommended in the Near-Term MPE. The Far-Term MPE is described through general recommendations of the configuration with supporting general operational concepts.

18.2.1 Cost Estimating

A software cost estimating capability needs to be developed for DMA. This capability has two aspects: (1) a methodology and (2) an automated tool; both of which need to be particularly adapted for DMA use. A research activity whose objective is the development of a software cost estimating capability would study in detail the technical software development and maintenance efforts, the management review procedures, and the working environment. The benefit of the research would be improved management control of costs and schedules.

18.2.2 Management

The management tools must also form an integral part of this methodology. This would require modification of available systems, or possibly their known method of utilization, or the development of DMA specific systems. The major problem with current tools is the level of users supported. In non-automated systems such as PERT a user may choose any or all parts of the methodology as applicable. In the automated systems all parts must be activated and few sections have default values, hence a large pre-utilization effort is required to establish metrics independent of the level of support required.

18.2.3 Tool Set Integration

The recommended tool set provides the state of the art in programmer aids. To further optimize programmer activity, an effort to integrate the tools and their interfaces should be undertaken.

18.2.4 Code Auditor

In order to enforce coding standards established by DMA, a research effort should be initiated for a coding auditor system. The purpose of such a system is to isolate portions of code which do not comply with established standards.

18.2.5 Ada

The Ada language has been developed to aid in the determination and implementation of new system development standards. It is envisioned that Ada will be used throughout the life cycle of all new systems from requirements specification through system maintenance. The specific problem is providing data on the use of Ada capabilities as applied to the life cycle methodology for large scale systems development. To implement a total life cycle methodology, the following tasks must be performed. First, information must be collected on the use of Ada in the design of large scale systems and on the training/skill levels needed to do this defined task. Next, this information must be utilized to create new development standards and associated curricula of instruction to train the work force. Finally, the new standards must be provided to the work force so that utilization of the standards and generation of the expertise needed at the systems specification and systems design levels will be achieved. This problem is a sigificant part of a much larger problem, the high cost of development/maintenance, and its solution will contribute a great deal to the advancement of system capabilities.

Additionally, due to the current status of Ada and its supporting environment, the adaptability of Ada to the DMA environment is not presently established. However, the desire of the Department of Defense to have a standard language and the strong features of the language indicate Ada will likely be used by DMA at some time in the future. Scenarios of Ada usage are another topic for further study.

19.0 CONCLUSIONS AND RECOMMENDATIONS

These recommended near-term and far-term environments meet the requirements as specified in the SON/SOC as well as provide for the environmental capabilities identified during the software tool evaluation. In the Near-Term MPE risks have been minimized by recommending tools which are currently available and have been throughly investigated with respect to claimed performance capabilities. Performance cannot be quantified, but cost data and rationale are provided which support our conclusions. An experimental system should be developed first in the implementation of the environment to provide engineering data to fine tune system performance. Additional data derived from this system will contribute to the development of tool usage methodologies, standards, and training programs during the inplementation of the Near-Term and Far-Term MPE's.

19.1 TRANSITION PLAN

The process of transitioning to the Near-Term and Far-Term MPE's must take into consideration DMA's capability to absorb the new technology without affecting the production environment.

19.1.1 Experimental Evaluation Systems

The initial Near-Term and Far-Term MPE systems at each DMA center will be developed and introduced as an experimental systems. One VAX should be acquired for delivery to the system developer and this system in turn will be used as a prototype for the experimental configurations. These configurations will then be evaluated on two VAX's resident at the DMA centers which will serve as test-beds for the developer's proposed methods of tool utilization. DMA will be required to select a group of personnel at each center to act as evaluators of the methodologies proposed by the developer within the DMA environment. A separate task of the group will be to participate in the development of materials and scenarios for use in training production personnel on the systems.

Near-Term and Far-Term full-scale production MPE system design will be parallel efforts to the experimental systems implementations and evaluations. To aid these efforts the transition plan has an underlying cycle in which as developer generates a methodology for utilization of a portion of either the Near-Term or Far-Term environments; the evaluation team analyzes the methodology within the DMA production environment; the developer incorporates changes as necessary;

and training materials and scenarios for the production Near-Term and Far-Term MPE's are upgraded. The following narrative presents the major tasks and milestones of the developer. A detailed schedule of the tasks and milestones is presented in Figure 19.1

In developing the cost benefit analysis for the DMA MPE, we identified the requirement to make tool sets available to DMA personnel during Phase I. Cost constraints for the Engineering Prototype development dictated that we only cost out one tool set at the contractor's facility. However, we recommend that two tool sets (one at each center), plus maintenance for two years, be included in Phase I costs to improve tool access by DMA personnel. This approach is necessary to avoid remote access problems caused by communication links to remote facilities. These problems were very detrimental to tool evaluations accomplished during the contract. This approach while not costed out in the reports, is certainly a more feasible way to accomplish prototyping at the DMAHTC and DMAAC sites. It is also required that these tool sets be hosted on VAX's that are identical to the contractor site. If this cannot be accomplished under current DMA procurements, we recommend that the two VAX's already identified to support Ada be added to the Phase I funding profile to ensure successful prototype implementation. These tool bearing hosts can still be used for Ada support during Phase IIA of the program.

```
MILESTONES/TASKS                  PROPOSED SCHEDULE BY QUARTERS
                                  Q1Q2Q3Q4Q1Q2Q3Q4Q1Q2Q3Q4Q1Q2Q3Q4Q1Q2
PHASE I. (1 VAX)                  #________________#     .       .
                                         .       .       .       .
o  DESIGN & IMPLEMENT                    .       .       .       .
     NEAR-TERM EXPERIMENTAL              .       .       .       .
     SYSTEM                       #________________#     .       .
o  DESIGN NEAR-TERM FULL-                .       .       .       .
     SCALE SYSTEM                        .#________#     .       .
o  TRAINING ON NEAR-TERM                 .       .       .       .
     EXPERIMENTAL SYSTEM              #____________#     .       .
o  DEVELOP METHODOLOGIES          #________________#     .       .
o  PRELIMINARY DESIGN OF                 .       .       .       .
     FAR-TERM EXPERIMENTAL               .       .       .       .
     SYSTEM                              .    #____#     .       .
                                         .       .       .       .
PHASE IA. (6 VAX'S)                      .       .#__________#   .
                                         .       .       .       .
o  IMPLEMENT FULL-SCALE                  .       .       .       .
     NEAR-TERM SYSTEM                    .       .#__________#   .
o  NETWORK VAX'S;                        .       .       .       .
     MAINFRAME LINK                      .       .     #_____#   .
o  TRAINING                              .       .#__________#   .
o  UPGRADE METHODOLOGIES                 .       .  #________#   .
     AND STANDARDS                       .       .       .       .
                                         .       .       .       .
PHASE II. (0 VAX'S)                      .       .    #________# .
                                         .       .       .       .
o  DESIGN & IMPLEMENT FAR-TERM           .       .       .       .
     EXPERIMENTAL SYSTEM                 .       .    #________# .
o  DESIGN FAR-TERM FULL-SCALE            .       .       .       .
     SYSTEM                              .       .       #_____# .
o  UPGRADE METHODOLOGY                   .       .       .  #__# .
o  IDENTIFY R&D EFFORTS                  .       .       .  #__# .
o  SOFTWARE TOOL INTEGRATION             .       .    #________# .
                                         .       .       .       .
PHASE IIA. (2 VAX'S)                     .       .       .    #______#
                                         .       .       .       .
o  IMPLEMENT FULL-SCALE SYSTEM           .       .       .    #______#
o  TRAINING                              .       .       .    #______#
o  UPGRADE METHODOLOGIES                 .       .       .       .
     AND STANDARDS                       .       .       .      #____#
```

Figure 19.1   Transition Schedule

19.1.2 Chronological Tasks

After delivery of the initial VAX to the system developer, the development of the VAX based tool utilization methodology begins. As the methodology is evolved for each tool a copy of the tool is distributed and hosted at each center on their experimental system. Once the methodologies for the tools have been developed, evaluated, and refined they must be integrated into a complete life cycle software development process. This process must then be merged with existing DMA standards. Specifically, a 4 volume set of standards is currently under development at DMA. Indications are that there will be little or no conflict between these standards and the recommended MPE, however, requirements for enhancements to the standards may be identified during the work with the experimental system. Near-Term MPE development will now be complete.

Implementation of the full-scale Near-Term production MPE systems will now proceed. Six VAX's are to be delivered two at a time (one to each center), during an estimated eighteen month period, to be used as software production computers. Additional implementation activities will include methodology/standards upgrades, production training, networking of Near-Term MPE VAX's within each center, and developing communication links between VAX's and the UNIVAC production mainframe.

Following the implementation of the full-scale Near-Term MPE environment work will begin on the development and evaluation of the Far-Term MPE. The developer's VAX will again be used as a prototype system to be used in the development of experimental systems at the DMA centers. However, the developer's original VAX computer will still be used to support training, engineering evaluation, and updates for the Near-Term production environment by downloading to the Near-Term environment as required from the evolving Far-Term system prototype. The original experimental VAX computers at each center will be phased from the Near-Term to Far-Term configurations. The development, implementation, and integration of the Far-Term MPE with the DMA standards will be similar to the Near-Term development. However, Far-Term development and implementation will benefit from the Near-Term transition experience. The resident Near-Term production systems can then be converted to the Far-Term MPE and the transition will be complete.

19.2 TRANSITION SUPPORT RECOMMENDATIONS

Two recommendations which evolved from the Tool Survey task are to provide long-term formal training to personnel prior to injecting new technology; and to train less experienced personnel within the production environment first. Additionally, MPE administrator and toolsmith functions would be beneficial in supporting the transition.

The MPE administrator and toolsmith functions would be support positions which would primarily serve as the focal point for management to observe the system activities and as an information source for MPE training. Specifically, the MPE administrator would be responsible for an overall understanding of the MPE and its use. The toolsmiths would aid the MPE administrator and would each be responsible for a thorough knowledge of a particular component of the MPE system. Personnel involved with the functions would be knowledgeable in the current tools and methodologies contained in the MPE as well as the VAX environment on which it would run. Tasks would include performing error rate studies, helping users with software development problems and the identification of needs not satisfied within the user/management communities. The personnel staffing this function should be located close to the MPE terminal areas in order to encourage programmers to bring their problems immediately instead of rerunning several times before giving up. The MPE administrator and toolsmith personnel would not be expected to debug user's programs but would be expected to help all users who had software development run problems using the MPE.

The transition plan provides for four months of training on each subset of the MPE. The recommendations concerning training, that it be long-term and with less experienced personnel first, are not the only recommendations which can be generated from the tool survey results. The major problem encountered during the evaluation task of the tool survey was schedule impacts due to access/hardware problems using remote equipment. This data strongly supports training DMA personnel on-site rather than using communication links to remote computers. Another problem was the training scenario/materials were not adequate due to problems of interpretation and development time. The transition plan provides for six months to develop a scenario and materials with DMA cooperation prior to training being accomplished on each software subset or system of the MPE. A final recommendation from the tool survey evaluation task is that the groups being trained be no larger than seven people to allow for proper interaction between instructor and group and amongst the group.

20.0 COST BENEFITS ANALYSIS

An important aspect of the Defense Mapping Agency Modern Programming Environment specification is its cost. This section contains the DMA MPE cost benefits analysis. The objectives of this cost benefits analysis are:

Objective 1: To estimate the total cost for the development and implementation of the DMA MPE as specified in this document.

Objective 2: To estimate the savings that can be realized by using the DMA MPE as compared to the continued use of existing DMA methods of software development and maintenance.

Objective 3: To predict the return on investment over a ten year time span starting from the beginning of MPE development.

To accomplish these objectives the following assumptions were made:

1) The MPE development and implementation task begins in July 1983 and ends in December 1987.

2) Hardware and software tool systems will be maintained by the manufacturers or vendors via maintenance agreements.

3) Government furnished software used in the MPE is available to DMA at essentially no cost.

In addition, the Defense Mapping Agency provided certain data particular to their organization. This data was:

1) Annual inflation rate of 5%,

2) Overtime burden rate of 1.9%, and

3) Typical DMA workyear cost of $25,000.

Figure 20.1 shows the major components in the cost analysis of the DMA MPE.

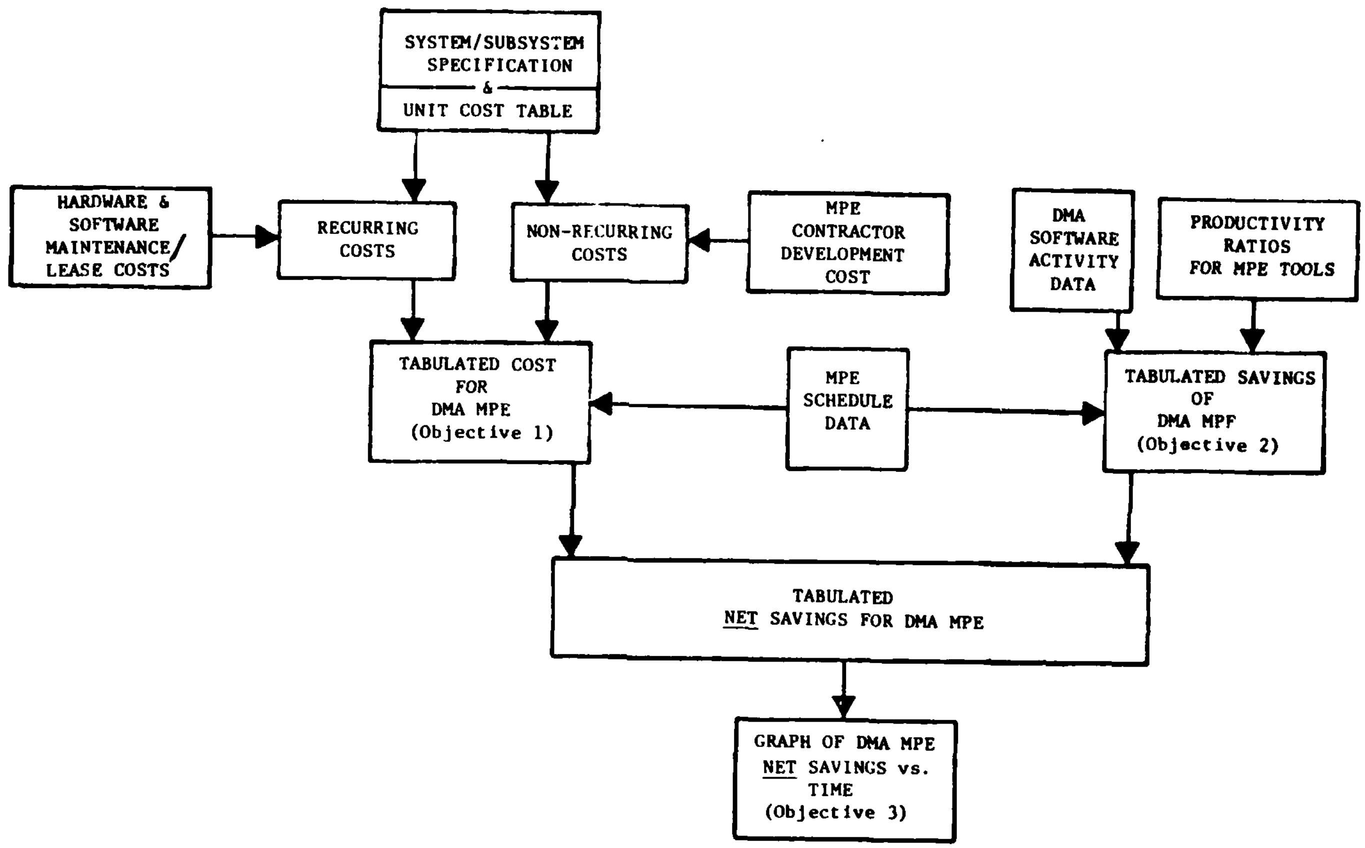


Figure 20.1 Block Diagram of DMA Modern Programming Environment Cost Analysis

For the purposes of cost estimating, Figure 20.2 shows the top-level configuration of the DMA MPE VAX computers. The solid lines indicate equipments that will be purchased or developed and hence their cost must be estimated. Those equipments indicated by dashed lines are already existing at the centers. The detailed cost of each particular hardware and software equipment item is shown in Figure 20.3, the unit cost table.

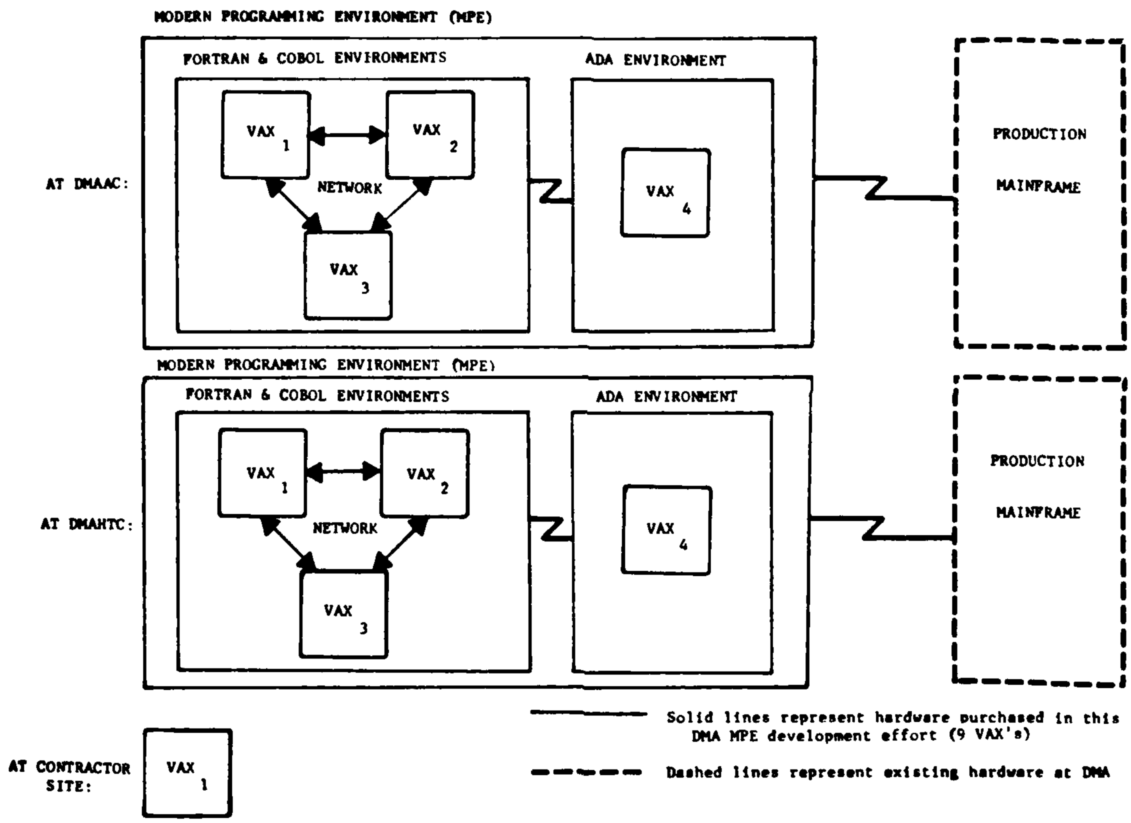


Figure 20.2 TOP LEVEL CONFIGURATION FOR DMA MPE VAX COMPUTERS

TOOL	HARDWARE/ SOURCE	OBJECT	MAINTENANCE	ANNUAL LEASE/RENT
USE.IT	- (11)	$142,000 (2)	$ 10,650 (1)	$ 2,500 (5)(8)
IS/1 PWB	$ 43,000	$ 31,500 (2)(4)	$ 7,500 (1)	- (7)
IS/1 INed	$ 30,000	$ 6,500 (2)	$ 1,250 (1)	- (7)
IS/1 INword	$ 20,000	$ 8,000 (2)	$ 1,500 (1)	- (7)
S.A.I. SDDL	$ 20,000	$ 5,000	$ 25 (12)	-
HYPERGRAPHICS	N/A	$ 500 (2)(4)	N/A	N/A
UNIX LICENSE	N/A	- (4)	N/A	N/A
VAX 11/780(6)	$274,900	N/A	$ 1,349 (8)	N/A
TERMINAL(9)	$ 2,700	N/A	$ 243 (1)	N/A
TERMINAL(10)	$ 2,400	N/A	$ 22 (8)	N/A
MULTITERMINAL EMULATOR	$ 8,100 (4)	N/A	INCLUDED	N/A
COMMUNICATIONS DEVICE	$ 1,575	N/A	$ 12 (8)	N/A
ASYNCHRONOUS MULTIPLEXERS	$ 6,500	N/A	$ 84 (8)	N/A
NETWORK SYSTEM	- (3)	N/A	$ 69,500 (1)(3)	$375,000 (1)(3)
FAVS	- (3)	N/A	INCLUDED	N/A
NETWORK LINK	$ 4,400	N/A	$ 39 (8)	N/A
PROTOCOL	- (13)	- (13)	- (13)	- (13)
DISK PACK	$ 1,500	N/A	N/A	N/A
TAPE(2400)	$ 30	N/A	N/A	N/A
VUE	$ 13,500	N/A	INCLUDED (1)	N/A

(1) DATA FOR FIRST YEAR ONLY
(2) SINGLE COMPUTER COST - DISCOUNTS AVAILABLE FOR ADDITIONAL LICENSES
(3) ALREADY RESIDENT AT DMA
(4) PERPETUAL LICENSE
(5) SHORT TERM
(6) INCLUDES O/S, CONSOLE, FLOPPY DISK, 2MB MEMORY, TAPE & DISK DRIVES
(7) MUST INCLUDE HARDWARE
(8) MONTHLY
(9) INtext II
(10) VT102
(11) SOURCE MAY BE PUT IN ESCROW
(12) PER UPDATE
(13) CAPABILITY EXISTS BUT MUST USE CUSTOMIZED SOFTWARE

Figure 20.3 Unit Cost Table

The data in this unit cost table represents the most current costs at the time of the preparation of this report. This data was aggregated into five cost estimating factors that were used to complete the cost estimate for the DMA MPE. These factors are:

1) Hardware purchases for one system	-$393K
2) Software purchases for one tool set	-$315K
3) Hardware maintenance costs for one system for one year	-$ 26K
4) Software maintenance costs for one tool setfor one year	-$ 27K
5) Average cost of one workyear of contractor labor	-$ 60K

To estimate the total cost for the DMA MPE, the first objective of the cost analysis, recurring and non-recurring costs were identified. Recurring costs included hardware maintenance, software maintenance, and personnel needed to directly support the MPE operation. Non-recurring costs included hardware purchases, software purchases, and contractor development labor. The non-recurring costs were confined to the the DMA MPE development time span (July 1983 to December 1987); however, the recurring costs with inflation included were distributed from July 1983 to December 1993 to be consistent with the return on investment estimate (objective 3). A tabulated cost for the DMA MPE was completed using this schedule data and the cost estimating factors. Figure 20.4 shows the total cost estimate for the DMA MPE by phase, funding source, and fiscal year. The total cost of the DMA MPE is approximately $11 million. Figure 20.5 shows how each fiscal year entry in Figure 20.4 is further partitioned into hardware, software, and labor. Hardware in this context means both hardware purchases and hardware maintenance, and similarly for software. Hardware costs approximately $4.2 million, software costs approximately $2.5 million, and contractor labor costs approximately $4.3 million.

PHASE	FUNDING TYPE	COSTS (in Thousands of Dollars)						TOTALS BY PHASE
		FY83	FY84	FY85	FY86	FY87	FY88	
I Near-Term Experimental System	R&D	90	1,210	690				1,990
IA Near-Term Full-Scale System	Production			1,490	3,630	120		5,240
II Far-Term Experimental System	R&D				770	730		1,500
IIA Far-Term Full-Scale System	Production					1,670	580	2,250
TOTALS BY FUNDING TYPE	R&D Production	90 -	1,210 -	690 1,490	770 3,630	730 1,790	- 580	3,490 7,490
TOTALS BY FISCAL YEAR	TOTAL	90	1,210	2,180	4,400	2,520	580	10,980

Figure 20.4 Cost Estimates for DMA MPE by Phase, Funding Source, and Fiscal Year

PHASE	RESOURCE	COSTS (in Thousands of Dollars)						TOTALS BY PHASE
		FY83	FY84	FY85	FY86	FY87	FY88	
I Near-Term Experimental System	Hardware		450	70				520
	Software		340	30				370
	Labor	90	420	590				1,100
	TOTAL	90	1,210	690				1,990
IA Near-Term Full-Scale System	Hardware			790	1,720	40		2,550
	Software			640	1,270	10		1,920
	Labor			60	640	70		770
	TOTAL			1,490	3,630	120		5,240
II Far-Term Experimental System	Hardware				110	40		150
	Software				40	10		50
	Labor				620	680		1,300
	TOTAL				770	730		1,500
IIA Far-Term Full-Scale System	Hardware					940	40	980
	Software					110	30	140
	Labor					620	510	1,130
	TOTAL					1,670	580	2,250
TOTALS BY RESOURCE	Hardware	-	450	860	1,830	1,020	40	4,200
	Software	-	340	670	1,310	130	30	2,480
	Labor	90	420	650	1,260	1,370	510	4,300
TOTALS BY FISCAL YEAR	TOTAL	90	1,210	2,180	4,400	2,520	580	$\boxed{10,980}$

Figure 20.5 Cost Estimates for DMA MPE by Phase, Resource, and Fiscal Year

The second cost analysis objective, the savings realized by using the DMA MPE, was completed by modeling the Defense Mapping Agency's usage of the Modern Programming Environment capabilities. First, data obtained from the General Dynamics questionnaire conducted in Stage 1 of this study was analyzed to estimate the percentage of total DMA software efforts that is spent in each software life cycle phase. Next, an estimate of the productivity improvement for each DMA MPE software tool was made. These two data were combined, and an estimated productivity improvement due to the DMA MPE capabilities of approximately 40% was calculated. Finally, the estimated size of the DMA programming population, an estimate of the rate of productivity improvement realization, and the DMA workyear costs with inflation were used to calculate an expected yearly dollar savings caused by the DMA MPE. The results of this calculation are shown in Figure 20.6.

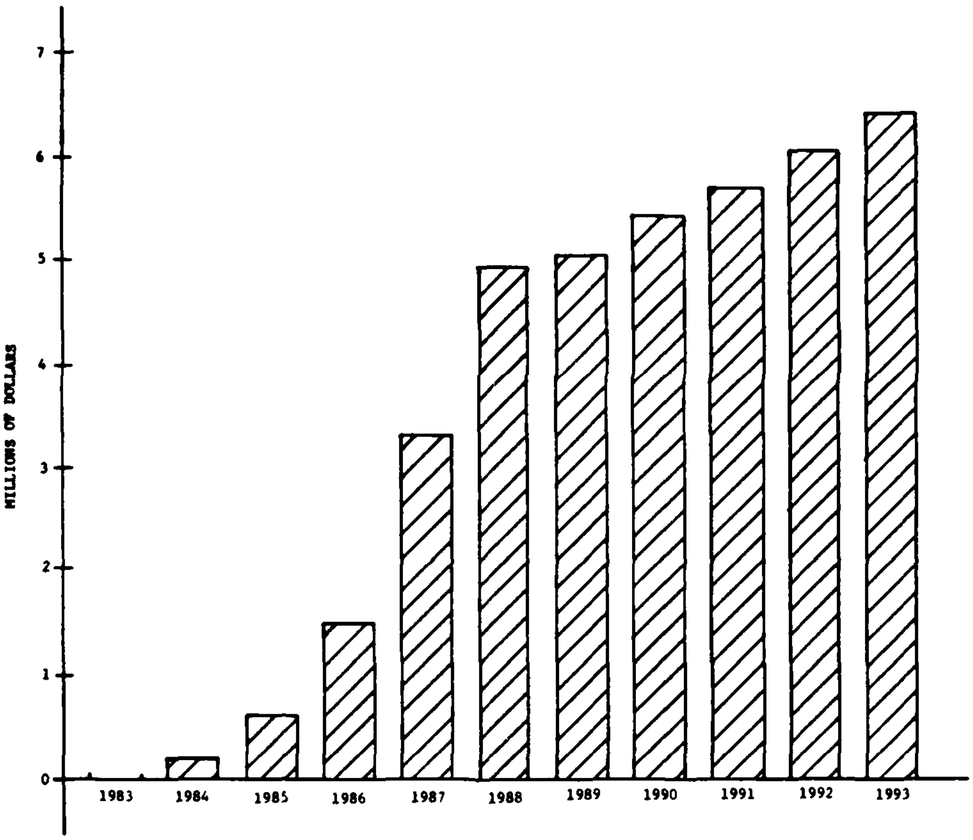


Figure 20.6 Yearly Savings of DMA MPE

To calculate the return on investment (objective 3) the tabulated yearly cost of the DMA MPE (the results from objective 1) was subtracted from the tabulated yearly savings of the DMA MPE (the results from objective 2). This net savings was accumulated and plotted in Figure 20.7. The cumulative net savings is the sum of the yearly savings minus yearly costs. Note that the total cost of the DMA MPE will be recovered after five years (in 1988), and after ten years (in 1993) an estimated cumulative net savings of $25 million will be realized. This represents an excellent return on the initial investment for the Defense Mapping Agency Modern Programming Environment.

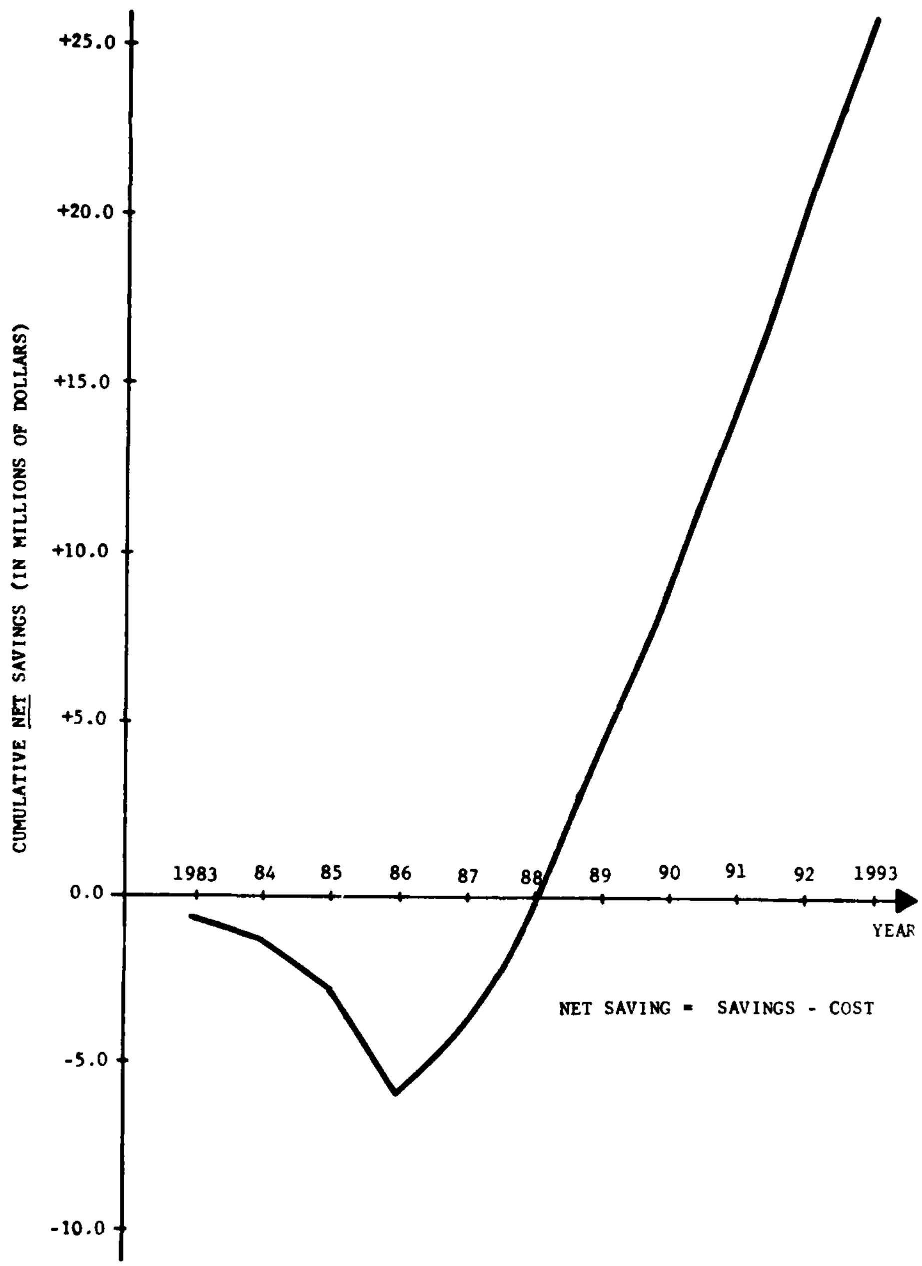


Figure 20.7 Graph of Cumulative Net Savings for DMA MPE Development

21.0 REFERENCES

1. FEDSIM (Federal Computer Performance Evaluation and Simulation Center) Installation Review - DMAHTC, November 1980

2. FEDSIM Installation Review - DMAAC, August 1980

3. FEDSIM Optimization and Error Rate Studies, February 1981

4. Statement of Operation Need and System Operational Concept, CDRL A002 for contract no. F30602-81-C-0039 - Interactive Computer Program Development System Study, February 1982

5. Tool Evaluation Plan, CDRL A003 for contract no. F30602-81-C-0039 - Interactive Computer Program Development System Study, September 1981

6. Tool Survey, CDRL A004 for contract no. F30602-81-C-0039, Interactive Computer Program Development System Study, February, 1982

22.0 LIST OF ABBREVIATIONS

ADP	AUTOMATED DATA PROCESSING
ADS	AUTOMATED DATA SYSTEM
AIAA	AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS
APSE	ADA PROGRAMMING SUPPORT ENVIRONMENT
ASCII	AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE
CDC	CONTROL DATA CORPORATION
CDRL	CONTRACT DATA REQUIREMENTS LIST
CIE	CONCEPT IMPLEMENTATION EVALUATION
CPU	CENTRAL PROCESSING UNIT
CRT	CATHODE RAY TUBE
DEC	DIGITAL EQUIPMENT CORPORATION
DLMS	DIGITAL LAND MASS SYSTEM
DMA	DEFENSE MAPPING AGENCY
DMAAC	DEFENSE MAPPING AGENCY AEROSPACE CENTER
DMAHQ	DEFENSE MAPPING AGENCY HEADQUARTERS
DMAHTC	DEFENSE MAPPING AGENCY HYDROGRAPHIC/ TOPOGRAPHIC CENTER
EDSC	EASTERN DATA SYSTEMS CENTER
FAME	FRONT END ANALYSIS AND MODELING ENVIRONMENT
FAVS	FORTRAN AUTOMATED VERIFICATION SYSTEM
FEDSIM	FEDERAL COMPUTER PERFORMANCE AND EVALUATION AND SIMULATION CENTER
GD/DSD	GENERAL DYNAMICS/DATA SYSTEMS DIVISION
HOL	HIGH ORDER LANGUAGE
HOS	HIGHER ORDER SOFTWARE
ICPDSS	INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY
IEEE	INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS
IPF	INTERACTIVE PROCESSING FACILITY
IPR	IN-PROCESS-REVIEW
KIPS	THOUSANDS OF INSTRUCTIONS PER SECOND
KOPS	THOUSANDS OF OPERATIONS PER SECOND
LAN	LOCAL AREA NETWORK
LARE	LOGICON'S AUTOMATED REQUIREMENTS ENGINEERING
MAPSE	MINIMAL ADA PROGRAMMING SUPPORT ENVIRONMENT
MEDS	MULTI-LEVEL EXPRESSION DESIGN SYSTEM
MPE	MODERN PROGRAMMING ENVIRONMENT
NBS	NATIONAL BUREAU OF STANDARDS
PRICE-S	PROGRAMMED REVIEW OF INFORMATION FOR COSTING AND EVALUATION - SOFTWARE
PSL	PROGRAM SUPPORT LIBRARY
PSL/PSA	PROBLEM STATEMENT LANGUAGE/PROBLEM STATEMENT ANALYZER
PWB	PROGRAMMER'S WORK BENCH
RADC	ROME AIR DEVELOPMENT CENTER
RAT	RESOURCE ALLOCATION TOOL
RSL/REVS	REQUIREMENTS STATEMENT LANGUAGE/REQUIREMENTS ENGINEERING

	VALIDATION SYSTEM
RJE	REMOTE JOB ENTRY
SCMS	SOFTWARE COMPLEXITY MEASUREMENT SYSTEM
SDDL	SOFTWARE DESIGN AND DOCUMENTATION LANGUAGE
SES	SOFTWARE ENGINEERING SYSTEM
SIP	SOFTWARE IMPROVEMENT PROGRAM
SON/SOC	STATEMENT OF OPERATION NEED & SYSTEM OPERATIONAL CONCEPT
SRIMP	SOFTWARE REQUIREMENTS INTEGRATED MODELING PROGRAM
TBD	TO BE DETERMINED
TBH	TOOL BEARING HOST
TEP	TOOL EVALUATION PLAN
TTY	TELETYPE
TX	TEXT EDITOR
UIFOLA	USER INTERFACE FOR ON LINE ASSISTANCE
UNCL	UNCLASSIFIED

23.0 APPENDICIES

This section contains the following 10 appendicies:

Appendix A DMA SURVEY QUESTIONNAIRE
Appendix B DEMONSTRATION RESPONSE FORM
Appendix C EVALUATION TOOL SET
Appendix D LIFE CYCLE QUESTIONNAIRES
Appendix E EVALUATION SURVEY RESPONSES SUMMARIZED
Appendix F EVALUATION ACTIVITY STATISTIS
Appendix G CONCEPT IMPLEMENTATION EVALUATION MATRIX
Appendix H SUMMARY STATISTICS FROM EVALUATION
    CIE BY NEED
    CIE BY TOOL
    CIE BY SCORE
    CIE BY CONCEPT
    BEST CASE BY NEED
    BEST CASE BY CONCEPT
    BEST CASE BY MODERN PROGRAMMING
        ENVIRONMENT FOR DMA
Appendix I LORAN NAVIGATIONAL LATTICE PROBLEM
Appendix J CONCEPT IMPLEMENTATION EVALUATION SHEETS

APPENDIX A
DMA SURVEY QUESTIONNAIRE

GENERAL DYNAMICS DMA SURVEY

Overview:

The Defense Mapping Agency (DMA) is involved in a study to design a modern programming environment computer system. To effectively produce this system, DMA must develop or obtain support programs/tools and accompanying procedures for product software development. Current projects frequently employ specialized support software packages unique to each development. It is evident that significant benefits can be realized if a common "core" of tools and procedures can be developed for all DMA centers.

The first phase of the investigation is a study of the needs of DMA and the tools which are available, or could be developed, to satisfy those needs. While the study is directed at production software, many of the tools being investigated have a wider potential and other areas such as technical support software will also be considered.

A questionnaire has been developed to assist in the study phase. It will help to determine needs at DMA and will help identify currently used tools appropriate for common use. This questionnaire will also function as a tool in validating the findings of the Boeing report, RADC-TR-79-343, as well as attempt to gather information about the future plans of DMA in the areas of operations and policies.

Your aid in identifying needs is appreciated and should lead to a system capable of supporting these needs in a cost effective manner. The questionnaire is only a beginning point for the study. Personal contacts will follow to clarify findings and to allow for additional inputs.

There are five parts to the questionnaire. Each person will be asked to answer three sections. The respondent section will be used to correlate answers with respect to a person's background. A tools section is included to gather general knowledge about what software tools are and their usefulness. One of three other sections will also be answered : 1) a technical section to gather data on operations 2) a management section to determine methods of operation 3) a policies section on DMA planning, control, organization and direction.

If you need additional space to respond to a question please attach extra sheets and indicate question section, number and letter as applicable.

NAME_______________________

DATE_______________________

ORGANIZATION_______________

PHONE NO.__________________

Answer each question as it pertains to software/hardware in your organization. When you don't feel qualified to respond to a question, please indicate this in the space left for comments. The comment space should also be used for any additional information that you feel is pertinent to the question. Please leave response areas blank if information is not known except when 'unknown' is an answer.

Organization DMAHQ_____ DMAHTC_____ DMAAC_____

1. Respondent characteristics:

A. Position description:___________________________________

B. Current project assignment:_____________________________

C. Total years experience in each category:(check correct range)

	Technical	Managerial	DMA
0 - 1/2 :	_____	_____	_____
1/2 - 1 1/2:	_____	_____	_____
1 1/2 - 3 :	_____	_____	_____
3 - 5 :	_____	_____	_____
Over 5 :	_____	_____	_____

D. Academic background:

	Field	Add. hours
____ Associate	___________________	__________
____ Bachelors	___________________	__________
____ Masters	___________________	__________
____ Doctoral	___________________	__________
____ Post Graduate	___________________	__________

E. List any OJT schools/seminars/classes attended:

____

____

____

F. List languages you are currently using:_________

____

____

____

G. List other languages with which you have had experience:_____________________________________

____

____

____

H. Computer(s) you are currently using:

Using	Years exp.
___________________	___________
___________________	___________
___________________	___________

I. Computer(s) you have used in the past:

Used	Years exp.
___________________	___________
___________________	___________
___________________	___________

J. Comments:_______________________________________

____

____

____

____

PART I: Technical

1. Project(s) Description. (This information should be general with respect to single/multiple projects being currently performed.)

   A. Indicate percentage of project(s) performed in following modes:

      _____ Interactive
      _____ Batch

   B. Estimate the percentage of work being performed and the responsible agency in the following life cycle(s). Percentages should only apply to DMA tasks.

	Percentage	DMA	Subcontracted
Requirements	__________	___	___
Design	__________	___	___
Coding	__________	___	___
Testing	__________	___	___
Maintenance	__________	___	___

   C. Computer(s) in use is __________________________

      Is reentrant code used for system processing(common banks)?

      Yes_____ No_____ Don't know_____

      Scheduling is accomplished _____automatically or _____manually.

      Indicate any of the following demand systems being used:

      _____ Conversational Time Sharing(CTS)
      _____ Editor(ED)
      _____ Symbolic Stream Generator(JCL)
      _____ Full Screen Editor(FSE)
      _____ Other __________________________

      What configuration management system for change control of production programs is in use? __________________________.

   D. Language(s) being used is _____________________

   E. On which medium does the application source program code(s) reside?

      Cassette _____
      Hardcopy _____
      Unknown _____
      Cards _____
      Tape _____

Disk _____

F. How is computer accessed?

Over-the-counter _____
Remote reader _____
Remote terminal _____

G. How is output received?

On-line application system _____
Tape _____
Remote printer _____
Microfilm _____
Over-the-counter _____
Remote terminal _____

H. How many technical personnel are involved with the task(s)? A range may be given for multiple projects. _____.

I. What is expected turnaround time in hours? _____

What is actual turnaround time in hours? _____

J. Fill in the blanks with characteristics which apply.

1. Require about_____K words of central memory.

2. Execution in approximately_____CPU seconds.

3. Requires_____(number of) secondary storage devices.
tape_____ disk _____ drum _____

4. Contains_____executable lines of code.

5. Documentation produced:

	Always	Often	Seldom	Never
Functional rqmts. desc.	______	_____	______	_____
Data rqmts. document	______	_____	______	_____
System/subsystem spec.	______	_____	______	_____
Program specification	______	_____	______	_____
Data base specification	______	_____	______	_____
User's manual	______	_____	______	_____
Test plan	______	_____	______	_____

Comments____
____

K. Software is generally developed

_____for a one-time application by a single user.
_____for application by a few users.
_____for application by many users (production software).
_____for application by few users, yet in reality is used by many.

Comments____
____

L. Development aids

	Used	Familiar	Name/comment
Tape Management System	_____	________	____________
MAP processors	_____	________	____________
Compilers	_____	________	____________
Assemblers	_____	________	____________
Linkage Editors	_____	________	____________
Text Editors	_____	________	____________
Configuration Control Aids	_____	________	____________
Other ____________________	_____	________	____________

M. Environment?

Do all qualified individuals have access to the computer(s)?_____
Is there a specialist staff assigned for computer access?_____

O. Security?

	Source	Data
Unclassified	_____	_____
Confidential	_____	_____
Secret	_____	_____
Top Secret	_____	_____

Is access to your physical area limited(e.g. SCI)?_______

P. Define the crew that will be working on your project(s):

Extensive experience - some top talent _____
Normal crew - experienced _____
Mixed experience - some new hire _____
Relatively inexperienced - many new hire _____

Q. How familiar is the project(s) to your organization?

Rework of previous project _____
Familiar type of project _____
Normal new project _____
No previous experience _____

R. Are there any factors that could complicate this project?

New hardware _____
New language _____
Single source development _____
Parallel to hardware development _____
Changing requirements _____
State-of-the-art advancement _____
Other_________________________________

S. Is a project generally integrated into a larger system? _____
If yes, is this a typical integration task for your organization? _____

T. Check area(s) which are typical of your project(s):

___ Editing
___ Word processing
___ Electronic mail/conferencing
___ Mathematical
___ String manipulation
___ Data storage and retrieval
___ Real-time command and control
___ Interactive operations
___ Operating system
___ Other _____________________

U. Comments?

____
____
____

2. Work Environment.

A. Is your work dependent upon previous work accomplished?_____
Is your work used by someone else when complete?_____

Comments____
____
____

B. Check if any item is used on your projects.

Notebook_____
Formal walkthroughs_____
Informal team sessions_____
Formal reviews_____
Training new personnel_____
Standards/guidelines_____
Backup person assigned for tasks_____
Trouble/problem reports_____
Configuration management_____
Coding style guide_____
Testing worksheets_____

C. Check the statements that apply to the way manhours, computer usage and schedule estimates are derived:

	Always	Often	Seldom	Never
1. formula(s)	______	_____	______	_____
2. project comparison	______	_____	______	_____
3. individual expertise	______	_____	______	_____
4. dictated by user	______	_____	______	_____
5. simulation	______	_____	______	_____
6. informally	______	_____	______	_____
7. Other	______	_____	______	_____

D. Check the statements that apply to resource planning:

	Always	Often	Seldom	Never
1. Guidelines used in plan preparation	______	_____	______	_____
2. Customer/user participation	______	_____	______	_____
3. Resources allocated by project	______	_____	______	_____
4. No formal planning (guesstimates)	______	_____	______	_____

E. Is the need for special skills/expertise recognized and addressed in the project plan?

_____Always _____Often _____Seldom _____Never

Comments____
____

F. Rank those items which are likely reasons for rewrite or change of the software requirements during the development effort. (1 = most likely, 4 = least likely).

_____ specification errors
_____ programmer errors
_____ ambiguous, incomplete or inconsistent specifications
_____ change in project requirements
_____ user and/or developer become better informed
_____ other:____

G. During the software planning process, are intermediate goals (project milestones) defined? _____YES _____NO

a. If YES, are these milestones used to monitor project progress?

_____Always _____Often _____Seldom _____Never

Comments____
____

H. Formal (documented) procedures to be followed during requirements specification

_____ do not exist
_____ exist and are followed
_____ exist but are not followed

I. The outcome of the requirements definition effort is:

	Always	Often	Seldom	Never
1. a formal, documented and approved requirements specification	______	_____	______	_____
2. an informal agreement with user/customer	______	_____	______	_____
3. a loosely defined set of requirements which is subject to change during project development	______	_____	______	_____
4. Other	______	_____	______	_____

J. The design effort is complete when:

	Always	Often	Seldom	Never
1. assurance is given that all requirements have been addressed	_____	_____	_____	_____
2. the scheduled due date for design completion is reached	_____	_____	_____	_____
3. the next-lower level of design would result in implementation decisions				
4. The user has reviewed and approved the design	_____	_____	_____	_____

K. Check any design techniques used.

_____ Structured coding
_____ Walkthroughs
_____ Peer reviews
_____ Top-down design
_____ Naming conventions
_____ Modular coding
_____ Data/File formatting
_____ Commenting conventions
_____ Design language
_____ Pseudo-code
_____ Flowcharts
_____ HIPO charts
_____ Other:____

L. From the list below, check the tools which are used during software development and indicate your opinion of their usefulness and availability (Y = yes, N = no).

USED		SUPPORTED	AVAILABLE	DOCUMENTED
___	SNOOPY	_____	_____	_____
___	PMD	_____	_____	_____
___	Dynamic Dump Routines	_____	_____	_____
___	FLAP	_____	_____	_____
___	INDEX	_____	_____	_____
___	Univ of MD Text Editor	_____	_____	_____
___	FILESCAN	_____	_____	_____
___	FORFLO	_____	_____	_____
___	TIDY	_____	_____	_____
___	SIP/PAR	_____	_____	_____
___	TIP/DMS	_____	_____	_____
___	REFORMATTER	_____	_____	_____

___ STRUCTRAN-1 _____ _____ _____
___ FAVS _____ _____ _____
___ PSL _____ _____ _____

List the operating system utilities (i.e., DOWNDATER, PFCK, UNDO, TDUMP, SORTSDF, FLIST, LABEL, VTRAN, etc.) most frequently used in your organization.

____
____

Comments____
____
____

M. The reporting mechanisms used within projects are:(indicate with a V for verbal or M for mechanical)

___ Weekly status form
___ Milestone charts
___ Technical memos
___ Informal meetings
___ Formal reviews
___ Notebook
___ Interactive mail
___ Other ______________________________________

Comments____
____
____

N. When is software documentation generally produced:

___ As the software is being developed
___ After the software has been developed
___ Only when required by the project plan

Comments____
____
____

O. At which levels and by whom do software testing and evaluation occur?

	DESIGNER	QA	USER
___ Module level	____	____	____
___ At module integration	____	____	____
___ At system testing	____	____	____
___ Other:______________	____	____	____

P. Does completed software undergo quality assurance testing by a group which is independent of the development team?

___Always ___Most of the time ___Sometimes ___Never

Comments____
____
____

Q. Are formal test plans/test strategies developed and documented?

___Always ___Most of the time ___Sometimes ___Never

Do these include testing to insure that all requirements have been met?

___Always ___Most of the time ___Sometimes ___Never

R. Do quality assurance procedures or guidelines exist?
___YES ___NO

Check the activities to which they apply and the degree to which they are followed:

		Rigidly	Nominally	Not used
___	Requirements specification	___	___	___
___	Design specification	___	___	___
___	Coding	___	___	___
___	Documentation	___	___	___
___	Testing	___	___	___
___	Maintenance	___	___	___
___	Redesign, code, retest, etc.	___	___	___

Comments____
____
____

S. During the program maintenance, estimate the percent of effort spent in each of the following:

___ Analysis and respecification of requirements
___ Redesign
___ Coding
___ Retesting (by developers)

Comments____
____
____

PART II: MANAGEMENT

1. Software Development

		Always	Often	Seldom	Never
A.	Do you develop software for varied hardware configurations?	______	_____	______	_____
	Do you encounter software transfer problems?	______	_____	______	_____
	Do you design software to be portable?	______	_____	______	_____

Comments____

____

____

B. Describe the frequency of errors discovered in operational software:

____ Very low ____ Low ____ Moderately low
____ Very high ____ High ____ Moderately high

Comments____

____

____

C. Estimate the percent of total effort devoted to

___ Adding new capabilities to existing programs
___ Starting from scratch to produce a new program
___ Detection/correction of errors in existing production programs

Comments____

____

____

D. Rank the following as used in project planning:
(1 = not used; 5 = often used)

___ Milestone identification
___ Resource availability
___ Manloading
___ Cost analysis
___ Schedule criticality

In which area could the most improvement be achieved?

____

E. To what extent is the user involved in the development effort?

____none ____some ____adequate ____too much

F. What types of reviews are used and with whom to track development?

	Management	Technicians
___ Walkthroughs	_____	_____
___ Design reviews	_____	_____
___ Informal meetings	_____	_____
___ Formal reviews	_____	_____
___ Status reports	_____	_____
___ Trouble reports	_____	_____
___ Technical memos	_____	_____
___ Other:________	_____	_____

G. Is development hindered by paperwork?_____ List reports commonly generated.___________________________________

____

H. What methods currently exist for documenting development?

___ Program design language
___ Programming style guides
___ Programming standards
___ Design specifications
___ User's manuals
___ Configuration controls
___ Other:____

Which methods are used? (Make second check mark)

2. Project Support

A. Rate the following support activities/roles in their effect upon project completion/failure (H = high influence; L = low influence):

___ Secretary
___ Keypunch
___ Technical consulting
___ Turnaround time
___ Machine access
___ Computer scheduling
___ Computer resource allocation
___ Paperwork
___ Reviews
___ Planning
___ Training support

B. Which are qualities of DMA software?

___ Portable
___ Modifiable
___ Understandable

C. Do guidelines exist to insure quality software develop ment?_____

Comments____

____

____

D. Rate the following as to their importance in software development. (1 = not important; 5 = very important)

___ Budget
___ Schedule
___ User involvement
___ Maintainability
___ Portability
___ Performance
___ Personnel productivity
___ Documentation
___ Other:____

E. What reports are received on project(s), how often are they received and are they verbal or mechanical?

Report	Frequency	Verbal/Mechanical
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____

3. Future Plans

A. What hardware do you expect to be replaced in the next five (5) years? Indicate approximate replacement date and new system to be installed.______________________________

____

____

B. Are there any new areas of support which you believe will be important in the next five (5) years?________________

____

C. What is the most important problem now facing your organization?____

D. What do you believe is the solution to increasing computer-enhanced productivity at DMA?______________________________

4. Current Projects

A. Are there any plans in progress or planned which would have a bearing on this study?___________________________________

B. Are there existing or planned DMA operational capabilities requiring additional analysis?______________________________

C. Comments:____

PART III: DIRECTION/POLICIES

A. How are work assignments made in your organization? __________
____
____
____

B. What are the long-term objectives of your organization?_______
____
____
____

C. Describe your overall plan to identify, evaluate and introduce new tools and techniques to DMA?_____________________________
____
____

D. What percent of your operational software was orginally developed by a subcontractor?_____

E. What percent of your software is maintained by
Subcontractor?______ DMA personnel?______

What percent of new software development will be done by
Subcontractor?______ DMA personnel?______

Is there a general policy regarding these allocations?
____
____
____

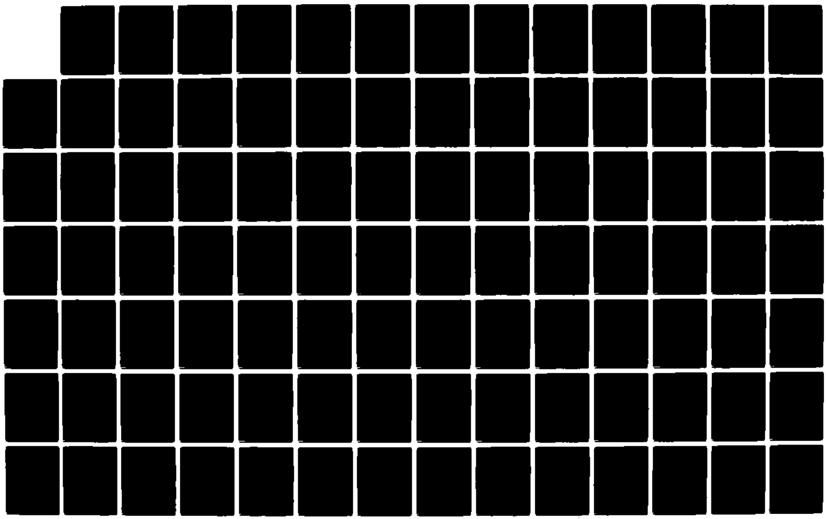
F. What are your long range plans for in-house training for modern programming environment practices?_________

G. How are you future software data processing requirements defined?____
____
____

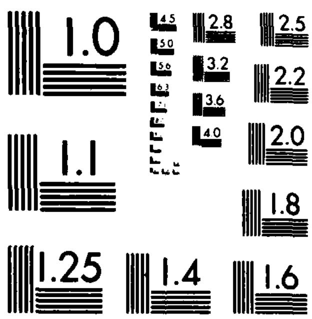
What levels of technical and management personnel are involved in future requirements planning?_______________________________
____

AD-A125 626 INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY VOLUME 1(U) GENERAL DYNAMICS FORT WORTH TX FORT WORTH DIV H C CONN ET AL. JAN 83 DMA-2-014-VOL-1 3/5

UNCLASSIFIED RADC-TR-83-3-VOL-1 F30602-81-C-0039 F/G 9/2 NL



M-2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

H. What reports are received on project(s), how often are they received and are they verbal or mechanical?

Report	Frequency	Verbal/Mechanical
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____
____________________	_________	_____/_____

I. Describe any current and/or projected capabilities/deficiences at DMA as they relate to adequately performing required programming functions. ______________________________________
____
____
____

J. Describe any existing or planned DMA operational capabilities requiring additional analysis? ______________________________
____

PART IV: TOOLS/COMMENTS

A. Check each area in which you use or could use a software tool to increase your productivity.
(u = currently used; c = could use).

REQUIREMENTS
___ Definition
___ Validation

DESIGN
___ Simulator
___ Program design language
___ Standardization

DEVELOPMENT
___ Compiler
___ MAP processor
___ Assembler
___ Linkage editor
___ Text editor
___ Configuration control
___ Security

TESTING
___ Test generator
___ Test validator

DOCUMENTATION
___ Flowchart generator
___ Automated text management system
___ Software documentation language
___ Graphics aids

MANAGEMENT
___ Cost estimating
___ Budget tracking
___ Report generator
___ Historical data base

OTHER
____
____
____

B. Do the tools/techniques in DMA organizations interface
____Well _____With effort _____Poorly.

C. Would a new tool be easily introduced into your DMA organization._____ If not, please explain why.________

D. Are there any general comments you would like to make concerning this questionnaire? Please indicate any area you believe we did not cover.

APPENDIX B
DEMONSTRATION RESPONSE FORM

DSS/DNA TOOL DEMONSTRATION EVALUATION

1. Name of tool being evaluated: ______________________________

2. Your name: ______________________________

3. Your organization: ______________________________

4. Your commercial telephone number: ______________________________

5. Your evaluation of the ease of input data preparation:

   _____ High, _____ Medium, _____ Low

6. Are there modifications to the input data preparation that would make the tool easier to use in the DNA environment?

   _____ Yes, _____ No

7. If yes, describe these modifications: ______________________________

   ____

8. Your evaluation of the ease of understanding the output results:

   _____ High, _____ Medium, _____ Low

9. Are there modifications to the output results format that would make the tool more useful in the DNA environment? _____ Yes, _____ No

10. If yes, describe these modifications: ______________________________

    ____

11. Do you perceive an application of the tool to DNA projects in the near-term (FY 1982)? _____ Yes, _____ No

12. If yes, which particular projects and how would you apply the tool to each project?

<u>Project</u>	<u>Application</u>
a. ____________	a. ____________
b. ____________	b. ____________

c. ____________________	c. ______________________________
d. ____________________	d. ______________________________

13. Do you perceive an application of this tool to DHA projects in the far term (FY 1985)? _____ Yes, _____ No

14. If yes, which particular projects and how would you apply the tool to each project?

Project	Application
a. ____________________	a. ______________________________
b. ____________________	b. ______________________________
c. ____________________	c. ______________________________
d. ____________________	d. ______________________________

15. Does this tool have functions that are also present in currently available DHA tools? _____ Yes, _____ No

16. If yes, what are these functions and in which currently available tool?

Function	Currently Available Tool
a. ____________________	a. ______________________________
b. ____________________	b. ______________________________
c. ____________________	c. ______________________________
d. ____________________	d. ______________________________

17. If both the demonstrated tool and the currently available tool have the same function, which one would you prefer to use and why

Function	Demonstrated Tool	Available Tool	Reason for Choice
a. ________	______________	______________	______________
b. ________	______________	______________	______________

APPENDIX C
EVALUATION TOOL SET

EVALUATION TOOL SET

The following set of tools was selected by the process explained in section 3.1.

FAME : Front-End Analysis & Modeling Environment

FAME is a microprocessor based system for interactively developing, analyzing and displaying Higher Order Software (HOS) and other system models in a user friendly environment. The nature of the HOS model is such, that when completed it can be the basis for projection to a variety of forms such as Structured Design Diagrams, SADT and IDEF Diagrams, Petri-Nets, Data Flow Diagrams, PSL/PSA Source Code,etc. The user's interface with the analyzer is easily recognized by any current user of a structured modeling approach; therefore extensive training is unnecessary. Futhermore, when all the system capabilities are used one can check on proper usage of data types, functions and control structures and thereby add a new dimension to the design process that will lead to better, and more easily verified software designs. System features include: prompted interactive model development; analysis of modeling errors; graphic output of models; and conversion programs to a number of standard methodologies.

FORMAT

FORMAT is a text processor which is a useful tool for anyone involved in producing documentation, reports, correspondence, or other written material. A text processor automatically does many of the tedious and time consuming chores needed to produce a finished product, such as right margin justification, page numbering, chapter and section numbering, centering, table of contents and index generation, and other similar operations.

IS/1 WORKBENCH FOR VAX

The IS/1 Workbench for the VAX is a facility that provides a convenient working environment and a uniform set of tools for computer program development, document preparation and text processing. It is a general-purpose, multi-user, interactive system based on Bell Laboratories' PWB/UNIX system specifically engineered to make the designer's, programmer's and documenter's environment simple, efficient, flexible and productive.

MAPPER

MAPPER is a real-time data base management system in the UNIVAC series 1100 environment. The software system is specifically designed to efficiently support the intense mix of activity inherent in the Real-Time Report Processing/Generating environment and still allow demand and batch background processing. The Series 1100 Operating System interfaces with MAPPER 1100 functions through the MAPPER Supervisor, which controls terminal polling, function loading and execution, and storage. Breakpoint and usage algorithms are established by which MAPPER Supervisor prioritizes all internal activity to minimize response time, giving highest priority to low impact activities.

NODAL

NODAL is an execution path flow analyzer designed to aid the user in executing all the source code and all the branches in testing a FORTRAN program. It uses the technique of analyzing the code that will record the execution of the program's nodes. At the normal end of an execution of the user's instrumented program, NODAL will obtain control and provide information about the frequency of execution of each node. Also provided is a test effectiveness ratio (nodes executed/nodes identified) for each routine, a test effectiveness ratio for the entire program, and a list of the program nodes not executed.

OPTIMA

The SPERRY UNIVAC 1100 Project Management System (OPTIMA 1100), an integrated system for project planning and control, is based on networking techniques. The OPTIMA 1100 System performs time analysis, cost analysis, resource analysis, resource allocation, report processing, including network plots, and maintenance/updating of OPTIMA 1100 mass storage files. The overall design is an integrated system comprising these functions.

SCMS : Software Complexity Measurement System

SCMS is an analysis tool that computes three types of complexity (Cyclomatic, Essential, and Actual) and graphically displays the control structure for each module of an input program. The complexity measure is based on a graph-theoretic approach to the analysis of programs developed by McCabe and provides information about how complicated (Cyclomatic Complexity), how well structured (Essential Complexity), and how well tested (Actual Complexity) a module is. This tool also provides a tree data structure that will identify modular interaction for the entire program.

SDDL : Software Design and Documentation Language

The objective of the Software Design and Documentation Language (SDDL) is to provide an effective communications medium to support the design and documentation of complex software applications. This objective is met by providing (1) a processor which can convert design specifications into an intelligible, informative machine-reproducible document, (2) a design and documentation language with forms and syntax that are simple, unrestrictive, and communicative, and (3) methodology for effective use of the language and processor. The processor has the capability to format documents, summarize design information in the form of reports and handle various user-controlled directives.

TX

The TX text editor is a stand-alone interactive program that is intended exclusively for full-screen, interactive text editing of ASCII files on Harris Models 2300, 8610, and 8680 CRT's.

APPENDIX D
LIFE CYCLE QUESTIONNAIRES

REQUIREMENTS SPECIFICATION EVALUATION CRITERIA

1. Are you familiar with any of the following requirements specification techniques? YES NO
   - PSL/PSA (University of Michigan) ___ ___
   - Structured Analysis (Yourdan) ___ ___
   - ISDOS (University of Michigan) ___ ___
   - CADSAT (U. S. Air Force) ___ ___
   - SREP (U. S. Army) ___ ___
   - SADT (SofTech) ___ ___
   - SAMM (Boeing) ___ ___
   - RLP (GTE) ___ ___
   - SREM (TRW) ___ ___
   - IDEF (SofTech) ___ ___
   - If yes, is your knowledge from
     general background ___
     formal training ___
     actual use ___

2. Is the requirements tool user friendly (that is)?
   - Is it easy to learn? ___ ___
   - Is it easy to use? ___ ___
   - Does it promote user satisfaction? ___ ___
   - Are error diagnostics understandable without recourse to study or documentation? ___ ___
   - Does it provide help facilities? ___ ___
   - Does it recognize different levels of users? ___ ___
     (That is - from novice to experienced)
     If yes, characterize the levels as you perceive them.

3. Does the tool ease the task of decomposing the problem into functions? ___ ___

4. Does the tool allow traceability between requirements and design? ___ ___

5. Does the tool identify inconsistencies in requirements? ___ ___

6. Does the tool promote a top-down approach? ___ ___

7. Is user documentation task reduced through the use of the tool? ___ ___

8. a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved requirements specification.) ____ ____

b. Were more specific requirements surfaced as a result of using this tool? ____ ____

9. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	____	____
- 2 to 5 hrs	____	____
- 5 to 10 hrs	____	____
- greater than 10 hrs	____	____

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

10. Is the level of detail for the problem statement sufficient to allow requirements specification? ____ ____

11. Does the tool do what it's advertised to do? ____ ____

12. Do you think this tool is applicable to the DMA environment? Explain. ____ ____

13. Provide your assessment of the tool and, if possible, compare this tool with other requirements tools/techniques with which you are familiar.

14. Succinctly describe the best feature(s) of this tool.

15. Succinctly describe the worst feature(s) of this tool.

15. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

DESIGN DEFINITION EVALUATION CRITERIA

		YES	NO
1.	Are you familiar with any of the following design techniques		
	- SADT	___	___
	- SDDL	___	___
	- JACKSON	___	___
	- HOS	___	___
	- HIPO	___	___
	- WARNIER	___	___
	- ORR	___	___
	- PETRI-NETS	___	___
	- If yes, is your knowledge from		
	general background ___		
	formal training ___		
	actual use ___		
2.	Is the DESIGN tool user friendly (that is)?		
	- Is it easy to learn?	___	___
	- Is it easy to use?	___	___
	- Does it promote user satisfaction?	___	___
	- Are error diagnostics understandable without recourse to study or documentation?	___	___
	- Does it provide help facilities?	___	___
	- Does it recognize different levels of users? (That is - from novice to experienced) If yes, characterize the levels as you perceive them.	___	___
3.	Does the tool ease the task of defining the problem functions?	___	___
4.	Does the tool allow traceability between requirements and design?	___	___
5.	Between design and coding?	___	___
6.	Does the tool promote modularity in design?	___	___
7.	Does the tool promote a top-down approach?	___	___
8.	Is user documentation task reduced through the use of the tool?	___	___
9.	Was the coding task shortened or eased in any way by the use of this tool?	___	___
10.	a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved design definition.	___	___
	b. Were more new requirements surfaced as a result of using this tool?	___	___

11. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	____	____
- 2 to 5 hrs	____	____
- 5 to 10 hrs	____	____
- greater than 10 hrs	____	____

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

12. Was the level of detail in the requirements statement sufficient to allow design definition? ____ ____

13. Does the tool do what it's advertised to do? ____ ____

14. Do you think this tool is applicable to the DMA environment? Explain. ____ ____

15. Provide your assessment of the tool and, if possible, compare this tool with other design tools/techniques with which you are familiar.

16. Succinctly describe the best feature(s) of this tool.

17. Succinctly describe the worst feature(s) of this tool.

18. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

CODING PHASE EVALUATION CRITERIA

TOOL EVALUATED: ____________________

1. Are you familiar with any of the following coding techniques or practices? YES ___ NO ___
   - FORTRAN Structured Programming Concepts ___ ___
   - Top-down Implementation ___ ___
   - Structured FORTRAN preprocessors ___ ___
   - Modularization Criteria ___ ___
   - COMMON Data Usage Guidelines ___ ___
   - Formal/Actual Subroutine Parameter Conventions ___ ___
   - Program Documentation Criteria ___ ___
   - Code Identification Guidelines ___ ___
   - If yes, is your knowledge from
     - general background ___
     - formal training ___
     - actual use ___

2. Is the coding tool user friendly (that is)?
   - Is it easy to learn? ___ ___
   - Is it easy to use? ___ ___
   - Does it promote user satisfaction? ___ ___
   - Are error diagnostics understandable without recourse to study or documentation? ___ ___
   - Does it provide help facilities? ___ ___
   - Does it recognize different levels of users? (That is - from novice to experienced) If yes, characterize the levels as you perceive them. ___ ___

3. Does the tool ease the task of coding the designed functions? ___ ___

4. Does the tool allow traceability between design and coding? ___ ___

5. Does the tool allow traceability between coding and testing? ___ ___

6. Does the tool promote modularity in coding? ___ ___

7. Does the tool promote a top-down approach? ___ ___

8. Is the user documentation task reduced through the use of the tool? ___ ___

9. Was the coding task shortened or eased in any way by the use of this tool? ___ ___

	YES	NO
10. a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved coding capabilities?)	____	____
b. Were design modifications identified as a result of using this tool?	____	____

11. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	____	____
- 2 to 5 hrs	____	____
- 5 to 10 hrs	____	____
- greater than 10 hrs	____	____

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

12. Was the level of detail in the design statement sufficient to allow code implementation?	____	____
13. Does the tool do what it's advertised to do?	____	____
14. Do you think this tool is applicable to the DMA environment? Explain.	____	____

15. Provide your assessment of the tool and, if possible, compare this tool with other coding tools/techniques with which you are familiar.

16. Succinctly describe the best feature(s) of this tool.

17. Succinctly describe the worst feature(s) of this tool.

18. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

TESTING PHASE EVALUATION CRITERIA

TOOL EVALUATED: ______________________

		YES	NO
1.	Are you familiar with any of the following testing tools?		
	- FAVS	___	___
	- NODAL	___	___
	- SCMS	___	___
	- RXVP80	___	___
	- DAVE	___	___
	- FASP	___	___
	- FORTRAN'77 ANALYZER	___	___
	- SOFTOOL 80	___	___
	- If yes, is your knowledge from general background ___ formal training ___ actual use ___		
2.	Is the testing tool user friendly (that is)?		
	- Is it easy to learn?	___	___
	- Is it easy to use?	___	___
	- Does it promote user satisfaction?	___	___
	- Are error diagnostics understandable without recourse to study or documentation?	___	___
	- Does it provide help facilities?	___	___
	- Does it recognize different levels of users? (That is - from novice to experienced) If yes, characterize the levels as you perceive them.	___	___
3.	Does the tool ease the task of testing the designed functions?	___	___
4.	Does the tool allow traceability between coding and testing?	___	___
5.	Does the tool promote modularity in testing?	___	___
6.	Is the user documentation task reduced through the use of the tool?	___	___
7.	Was the testing task shortened or eased in any way by the use of this tool?	___	___

	YES	NO
8. a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved testing capabilities?)	____	____
b. Were design modifications identified as a result of using this tool?	____	____

9. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	____	____
- 2 to 5 hrs	____	____
- 5 to 10 hrs	____	____
- greater than 10 hrs	____	____

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

	YES	NO
10. Was the level of detail in the design statement sufficient to allow comprehensive testing?	____	____
11. Does the tool do what it's advertised to do?	____	____
12. Do you think this tool is applicable to the DMA environment? Explain.	____	____

13. Provide your assessment of the tool and, if possible, compare this tool with other coding tools/techniques with which you are familiar.

14. Succinctly describe the best feature(s) of this tool.

15. Succinctly describe the worst feature(s) of this tool.

16. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

APPENDIX E
EVALUATION SURVEY RESPONSES SUMMARIZED

DMA TEAM PARTICIPANTS

DMAAC

Geopositional Department Photogrammetric Control Division	Robert Spors
Aerospace Cartography Department Cartographic Data Division	Norbert Pink
Scientific Data Department Scientific Computer Division	Larry Holmgren R. Dwane Kindsfather Charles Masback Billy Rice

DMAHTC

Geodesy Department Satellite Geophysics Division	Peter Mayer
Topography Department Techniques & Programming Division	Johnnie Bishop
Computer Services Department Scientific Computing Division	Mary Albert Mike Lewis Geri Loughney
Computer Services Department Techniques Office	Martha Plemmons Thomas P. Williams

1) In the following the numeric data is represented as AC/HTC.
2) Ambiguous answers such as "fairly", "somewhat", and "so-so" were annotated as a "yes" and a "no".
3) Sentences may be paraphrased to express main content.
4) If ranges were given, upper limit was used.
5) Comments are prefaced with "HTC" or "AC".

REQUIREMENTS SPECIFICATION EVALUATION CRITERIA
(BOTH CENTERS USED SAME)

	YES	NO
1. Are you familiar with any of the following requirements specification techniques?		
- PSL/PSA (University of Michigan)	2/1	4/5
- Structured Analysis (Yourdan)	2/1	4/5
- ISDOS (University of Michigan)	0/0	6/6
- CADSAT (U. S. Air Force)	0/0	6/6
- SREP (U. S. Army)	1/0	5/6
- SADT (SofTech)	1/1	5/5
- SAMM (Boeing)	0/0	6/6
- RLP (GTE)	0/0	6/6
- SREM (TRW)	0/0	6/6
- IDEF (SofTech)	0/0	5/6

- If yes, is your knowledge from
  - general background 1/1
  - formal training 1/1
  - actual use 0/0

	YES	NO
2. Is the requirements tool user friendly (that is)?		
- Is it easy to learn?	4/4	3/2

HTC - provided good quality formal training available.

	YES	NO
- Is it easy to use?	4/6	3/1
- Does it promote user satisfaction?	3/3	3/3
- Are error diagnostics understandable without recourse to study or documentation?	0/1	6/5
- Does it provide help facilities?	3/5	2/2

HTC - help facilities aren't helpful.

	YES	NO
- Does it recognize different levels of users? (That is - from novice to experienced) If yes, characterize the levels as you perceive them.	0/1	6/3
3. Does the tool ease the task of decomposing the problem into functions?	4/6	3/0

4. Does the tool allow traceability between

requirements and design? | 4/5 | 3/2

5. Does the tool identify inconsistencies in requirements? | 5/6 | 3/0

6. Does the tool promote a top-down approach? | 5/6 | 2/0

7. Is user documentation task reduced through the use of the tool? | 4/4 | 3/1

8. a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved requirements specification.) | 4/3 | 3/1

b. Were more specific requirements surfaced as a result of using this tool? | 3/4 | 3/1

9. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	2/1	2/1
- 2 to 5 hrs	3/0	0/1
- 5 to 10 hrs	1/2	2/2
- greater than 10 hrs	0/2	1/2

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

AC - 2, 5, 1, 2, 1
AC - Does not fit current procedures or policies; requires decisions by Sr. Analysts that they are not in a position to make.
HTC - 10, 10, 10, 1, 3

10. Is the level of detail for the problem statement sufficient to allow requirements specification? | 2/6 | 4/0

11. Does the tool do what it's advertised to do? | 3/4 | 0/2

12. Do you think this tool is applicable to the DMA environment? Explain. | 4/5 | 3/1

AC - Could be if context and use were well thought out
AC - Yes, forces top-down approach
AC - No
AC - Requires designer to think through specification
HTC - There is a need for better documentation and more detailed initial requirements

HTC - Few people are requested to handle a project from beginning to end.
HTC - Would help define all requirements of a program in the beginning eliminating wasted time in recoding problem.
HTC - Extensive amount of software and rapid programmer turnover makes this tool useful to DMA
HTC - No, too many bugs.

13. Provide your assessment of the tool and, if possible, compare this tool with other requirements tools/techniques with which you are familiar.

AC - Requires too much data and information from user.
AC - Good and useful.
AC - Poor.
AC - Gets you started with an overall view; organizes the problem.
HTC - Good tool; would be helpful in maintenance; high quality formal training needed.
HTC - Unnecessary.
HTC - Very good; reduces time to find solution; eliminates wasted programming time; shortens time to get program into production.
HTC - Has possibilities but too new, i.e., bugs.

14. Succinctly describe the best feature(s) of this tool.

AC - Computer produced tree.
AC - Graphic tree diagram; parameter checking; parent-offspring diagram; excellent documentation produced.
AC - Error checking; documentation of flow processes
HTC - Output is good documentation; tree diagram is useful in coding; prompts are helpful; fairly easy to use once learned.
HTC - Analysis.
HTC - Top-down functional decomposition.
HTC - Analysis; aid in organization and logic; good documentation for maintenance; aids in problem solution; catches errors early.
HTC - Tree diagram.

15. Succinctly describe the worst feature(s) of this tool.

AC - Requires too much data.
AC - Sometimes very complex.
AC - Error messages are cryptic.
AC - User documentation.

HTC - Errors hard to correct; many bugs; only allows single-step update.
HTC - Bugs in software.
HTC - Errors hard to correct; bad error messages; problem with global update; limited number of variables.

16. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

AC - Major problem faced was bad data communications line requiring repetitive work by team members.
AC - The syntax of the tool could distract from the original requirements.
AC - Could be used in design, but inappropriate for requirements.
HTC - Poorly documented; for different _options_, different _types_ of answers were required, i.e., numeric or alphanumeric and sometimes it was irrelevant; learning tool would require much use; tool is redundant to work accomplished; meaningless error messages given; correcting errors is difficult.
HTC - Best feature is analysis; diagnostics _not_ user friendly; only allows one update of a specific group at a time; multiple errors in documentation; A-J terminal has nice qualities but difficult to use.
HTC - Poor documentation.
HTC - Project too simple; errors in Fame software; poor user's manual.

DESIGN DEFINITION EVALUATION CRITERIA
(AC USED SDDL, HTC USED FAME)

1. Are you familiar with any of the following design techniques

	YES	NO
- SADT	0/1	6/5
- SDDL	1/1	5/5
- JACKSON	2/0	4/6
- HOS	1/0	5/6
- HIPO	2/2	4/4
- WARNIER	2/1	4/5
- ORR	0/1	6/5
- PETRI-NETS	1/0	5/6

- If yes, is your knowledge from
  - general background _2/1_
  - formal training _0/1_
  - actual use _0/0_

2.	Is the DESIGN tool user friendly (that is)?		
	- Is it easy to learn?	4/4	2/2
	- Is it easy to use?	4/6	2/1
	- Does it promote user satisfaction?	2/3	5/3
	- Are error diagnostics understandable without recourse to study or documentation?	1/1	5/5
	- Does it provide help facilities?	2/5	4/1
	- Does it recognize different levels of users? (That is - from novice to experienced) If yes, characterize the levels as you perceive them.	1/1	6/4
3.	Does the tool ease the task of defining the problem functions?	2/6	4/0
4.	Does the tool allow traceability between requirements and design?	4/5	2/2
5.	Between design and coding?	5/5	1/1
6.	Does the tool promote modularity in design?	4/6	3/0
7.	Does the tool promote a top-down approach?	5/6	1/1
8.	Is user documentation task reduced through the use of the tool?	4/6	2/0
9.	Was the coding task shortened or eased in any way by the use of this tool?	2/5	4/1
10.	a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved design definition.	3/4	4/1
	b. Were more new requirements surfaced as a result of using this tool?	3/5	3/3

11. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	5/0	1/0

- 2 to 5 hrs _1/0_ _1/2_
- 5 to 10 hrs _0/3_ _1/2_
- greater than 10 hrs _0/2_ _2/1_

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

SDDL:

AC - 1, 3

FAME:

HTC - 1, 1, 2, 2, 3, 5

12. Was the level of detail in the requirements statement sufficient to allow design definition? _3/6_ _3/0_

13. Does the tool do what it's advertised to do? _2/5_ _2/2_

14. Do you think this tool is applicable to the DMA environment? Explain. _4/5_ _3/1_

SDDL:

AC - Allows design in pseudo code.
AC - Could be used in program design phase.
AC - Seems like you are coding the program twice.

FAME:

HTC - Project is better understood; provides good documentation.
HTC - Shortens time for generating a program
HTC - Yes, but each programmer will have his own interface
HTC - Too many bugs still exist.

15. Provide your assessment of the tool and, if possible, compare this tool with other design tools/techniques with which you are familiar.

SDDL:

AC - Unwieldly to use.
AC - Of little value.
AC - Good to aid in maintenance
AC - Largely a pretty-printer.
AC - No, DMA does not work in an interactive environment.

FAME:

HTC - Fairly good, but emphasis should be on methodology rather than interaction with computer.
HTC - Limitations of tool creates problems in decomposition.
HTC - Too immature at this time.

16. Succinctly describe the best feature(s) of this tool.

SDDL:

AC - Formatted source listing; allows definition of control structures
AC - Module invocation tree; cross reference tables.
AC - Requires analyst to discipline his thinking.
AC - It does cross checking that a human author would not do.

FAME:

HTC - Capability to define I/O.
HTC - Output is good documentation, tree diagram is useful in creation of design and coding; prompts helpful; fairly easy to use once learned.

17. Succinctly describe the worst feature(s) of this tool.

SDDL:

AC - No parameter checking
AC - Flags required for cross references
AC - Too inflexible
AC - Doesn't do anything but a little editing
AC - Amount of information required for tool to perform.

FAME:

HTC - Typos not easy to correct and may cause errors which are unaccessible; unneeded prompts given; limited number of inputs; no convenient way to stop analysis; bug in global update requests; error messages weren't easily understood.
HTC - Expansion limitations; analysis limitations; variable limitations.
HTC - Limitations: size and number; errors hard to correct; couldn't access some inputs; problems with analysis.

18. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

SDDL:

AC - Would not be used at DMA without management pressure.

FAME:

HTC - A lot of time should be spent on formal training; multiple problems should be solved in training.
HTC - More troubles due to greater detail required.

CODING PHASE EVALUATION CRITERIA
(AC USED TX, HTC USED IS/1)

1. Are you familiar with any of the following coding techniques or practices?	YES	NO
- FORTRAN Structured Programming Concepts	6/5	0/1
- Top-down Implementation	5/4	1/2
- Structured FORTRAN preprocessors	3/2	2/4
- Modularization Criteria	4/2	1/4
- COMMON Data Usage Guidelines	4/5	1/1

- Formal/Actual Subroutine Parameter Conventions — 5/4 — 1/2
- Program Documentation Criteria — 5/2 — 1/4
- Code Identification Guidelines — 3/1 — 1/5
- If yes, is your knowledge from
  - general background 4/4
  - formal training 5/3
  - actual use 4/3

2. Is the coding tool user friendly (that is)?
   - Is it easy to learn? — 5/6 — 1/0
   - Is it easy to use? — 5/6 — 1/0
   - Does it promote user satisfaction? — 4/6 — 2/0
   - Are error diagnostics understandable without recourse to study or documentation? — 2/1 — 5/5
   - Does it provide help facilities? — 3/5 — 3/1
   - Does it recognize different levels of users? (That is - from novice to experienced) If yes, characterize the levels as you perceive them. — 1/1 — 5/4

3. Does the tool ease the task of coding the designed functions? — 3/4 — 2/2

4. Does the tool allow traceability between design and coding? — 2/3 — 3/3

5. Does the tool allow traceability between coding and testing? — 0/3 — 3/1

6. Does the tool promote modularity in coding? — 1/3 — 3/3

7. Does the tool promote a top-down approach? — 0/3 — 3/3

8. Is the user documentation task reduced through the use of the tool? — 1/2 — 3/4

9. Was the coding task shortened or eased in any way by the use of this tool? — 3/4 — 2/2

10. a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved coding capabilities?) — 2/3 — 3/3

    b. Were design modifications identified as a result of using this tool? — 0/5 — 4/2

11. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	4/1	1/0
- 2 to 5 hrs	0/1	2/2
- 5 to 10 hrs	0/2	0/1
- greater than 10 hrs	0/2	1/3

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

TX:

AC - 3, 5, 1, 1

IS/1:

HTC - 5, 2, 1, 2, 4

12. Was the level of detail in the design statement sufficient to allow code implementation? 4/6 1/0

13. Does the tool do what it's advertised to do? 3/5 0/0

14. Do you think this tool is applicable to the DMA environment? Explain. 3/4 2/2

TX:

AC - Only if DMA goes interactive.
AC - No, DMA is batch oriented.
AC - UNIVAC editor better.
AC - If implemented on DEC equipment.

IS/1:

HTC - Good data entry method.
HTC - UNIVAC is better.
HTC - Easy submission and documentation of code.

15. Provide your assessment of the tool and, if possible, compare this tool with other coding tools/techniques with which you are familiar.

TX:

AC - A great deal better than cards.
AC - Response too slow.
AC - Difficult to use and less powerful than UNIVAC.
AC - Powerful and useful; need special function keys.

IS/1:

HTC - UNIVAC and IS/1 redundant.
HTC - Access is a problem.
HTC - UNIVAC just as effective except for lack of terminals.
HTC - UNIVAC is equal to or better.
HTC - More familiar with MD editor and prefer it and

UNIVAC editors.

16. Succinctly describe the best feature(s) of this tool.

TX:

- AC - Context editing.
- AC - Full page capability.
- AC - Full screen editing; recovery capability; documentation.

IS/1:

- HTC - Library of object files.
- HTC - Text editing functions.
- HTC - Detachable keyboard; easy to learn; advanced editing system.
- HTC - Easy to use at low level.
- HTC - Editor facilities; easy to learn and use; documentation good.
- HTC - Easy to learn and use.

17. Succinctly describe the worst feature(s) of this tool.

TX:

- AC - Not user friendly; antagonistic.
- AC - Difficult to find line numbers; string change difficult to use.
- AC - Need special terminal.

IS/1:

- HTC - Text editor.
- HTC - Error messages not clear; hardware problems: Line noise and printer.
- HTC - Inconsistent access to files; diagnostic messages; cursor control only allowed in edit.

18. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

TX:

- AC - DMA must be interactive for it to be useful. Study showed that DMA programming environment is in stone-age.
- AC - Good text editors are important to coding.

IS/1:

- HTC - Documentation not up-to-date; access sometimes difficult; software bugs in terminal; error messages not friendly or not documented; the terminal hardware is one of the best, especially the keyboard.
- HTC - Would probably have liked more if more time for training had been available.

<u>TESTING PHASE EVALUATION CRITERIA</u>
(NODAL)

		YES	NO
1.	Are you familiar with any of the following testing tools?		
	- FAVS	4/1	1/6
	- MODAL	3/0	2/6
	- SCMS	0/0	3/6
	- RXVP80	1/0	3/6
	- DAVE	0/0	3/6
	- FASP	0/0	3/6
	- FORTRAN'77 ANALYZER	0/0	3/6
	- SOFTOOL 80	0/0	3/6
	- If yes, is your knowledge from		
	general background 3/0		
	formal training 2/1		
	actual use 3/1		
2.	Is the testing tool user friendly (that is)?		
	- Is it easy to learn?	0/0	4/0
	- Is it easy to use?	0/0	4/0
	- Does it promote user satisfaction?	0/0	4/0
	- Are error diagnostics understandable without recourse to study or documentation?	0/0	4/0
	- Does it provide help facilities?	0/0	4/0
	- Does it recognize different levels of users? (That is - from novice to experienced) If yes, characterize the levels as you perceive them.	0/0	4/0
3.	Does the tool ease the task of testing the designed functions?	1/1	2/0
4.	Does the tool allow traceability between coding and testing?	1/1	2/0
5.	Does the tool promote modularity in testing?	0/1	3/0
6.	Is the user documentation task reduced through the use of the tool?	0/1	3/0
7.	Was the testing task shortened or eased in any way by the use of this tool?	1/1	2/0
8.	a. Is productivity increased through use of the tool? (That is - is there a reduction in errors attributed to improved testing capabilities?)	1/1	2/0
	b. Were design modifications identified as a result		

of using this tool? _0/0_ _3/0_

9. How much time was spent in training (formal and on-the-job)?

	FORMAL	OJT
- 1 to 2 hrs	3/0	2/0
- 2 to 5 hrs	0/0	1/0
- 5 to 10 hrs	0/0	0/0
- greater than 10 hrs	-----	0/0

How much time is appropriate for other DMA users to learn to use this tool? (Express in terms of work days.)

AC - 5, 2

10. Was the level of detail in the design statement sufficient to allow comprehensive testing? _1/0_ _2/0_

11. Does the tool do what it's advertised to do? _1/0_ _0/0_

12. Do you think this tool is applicable to the DMA environment? Explain. _0/2_ _3/0_

HTC - Seems good.
AC - Not as good as FAVS.

13. Provide your assessment of the tool and, if possible, compare this tool with other coding tools/techniques with which you are familiar.

AC - Very limited as compared to other tools.
AC - All programming in FORTRAN '77, hence not applicable to DMA.
HTC - Sounds good.

14. Succinctly describe the best feature(s) of this tool.

AC - Allows dynamic testing.

15. Succinctly describe the worst feature(s) of this tool.

AC - Input options not user friendly.

16. Provide any comments about this tool and this portion of the evaluation which were not surfaced by the foregoing questions.

MAPPER

Used as a management tool to collect statistical data.

HTC - Extremely user friendly.
HTC - Knowing where to put SOE is complicated; summing two reports is complicated; user documentation could be reduced through use of the tool; UTS 400 is better than any terminals currently at DMA.
HTC - Good for report generating; missing capability of combining RIDs arithmetically.
HTC - Does not auto update on exit; keyboard excellent.
HTC - Nice keyboard; easy to use; easy to learn.
HTC - Keyboard a little confusing; multiple report combinations complicated.

GENERAL COMMENTS

These comments were received during discussions held at the end of the evaluations at each center. Comments which duplicated information in the prior section were not included.

AC - A comment was made that there was not enough time to do the evaluations. Discussion followed citing that the biggest impact on schedule and available time was access/hardware problems. Getting and maintaining phone lines to the remote computer sites was the major factor. Problems were encountered getting a line out of DMAAC, having a line/port available at the computer site, communicating with line noise present and physical phone availability. Support hardware also affected schedule. The first terminal delivered to support access to the Software Engineering System (SES) at GD/DSD in Fort Worth, Texas was non-operable when received. A second terminal delivered did work, but the mode of operation was slow (300 baud) due to SES communications capabilities. Additionally, to support the tools being used, specifically TX, a new protocol had to be set-up in the SES software causing an additional delay in access. Numerous times during the evaluation the SES was down for scheduled maintenance or broken during prime time business hours. A final scheduling problem was access to tools through only one terminal. This required the sequencing of all activities when many could have been performed in parallel.

Other comments included statements that the statistics to be gathered and the questionnaires to be answered could have been of better quality, as far as content and applicability. No suggestions were given for improvement. A final discussion centered on the size of the group involved in the effort. A general agreement was that the group size for this type of training should not be larger than seven.

HTC - The test-bed problem was considered to entail too much coding in proportion to other tasks involved in the life-cycle development. The terminals were well liked but more time should have been available to learn the intricacies of their use. This type of effort was considered a good learning technique, but more example problems should be covered, rather than one large problem.

EVALUATION QUESTIONNAIRE RESPONSE TRENDS

Life-cycle Covered by Questionnaire	Answers Yes/No AC	HTC	TOTAL	Comments Positive/Negative AC	HTC	TOTAL
Requirements Questions 2-8, 10-12	52/50	69/24	121/74	7/11	10/16	17/27
Design Questions 2-10, 12-14	61/56	84/28	117/84	7/10	4/7	11/17
Coding Questions 2-10, 12-14	42/46	70/35	112/81	6/7	9/10	15/17
Testing Questions 2-8, 10-12	6/41	6/0	12/41	1/4	2/0	3/4

APPENDIX F
EVALUATION ACTIVITY STATISTICS

DMAAC

TOOL	LIFE-CYCLE	TRAINING LABOR HOURS	COMP. HOURS	TOOL RUNS	USAGE ERRORS	T-BED DEVELOPMENT LABOR HOURS	COMP. HOURS	TOOL RUNS	USAGE ERRORS	T-BED ERRORS
FAME	REQUIREMENTS	45	4	7	7	79	19	9	16	17
SDDL	DESIGN	35	8	12	7	100	32	27	7	19
TX	CODING	15	5	3	9	128	65	111	45	18
NODAL	TESTING	4	0	0	8	17	3	0	0	0
MAPPER	DATA BASE	8	2	5	0	5	6	12	4	0
OPTIMA	MANAGEMENT	0	0	0	0	0	0	0	0	0
FORMAT	DOCUMENTATION	8	0	0	0	0	0	0	0	0

DMAHTC

TOOL	LIFE-CYCLE	TRAINING LABOR HOURS	COMP. HOURS	TOOL RUNS	USAGE ERRORS	T-BED DEVELOPMENT LABOR HOURS	COMP. HOURS	TOOL RUNS	USAGE ERRORS	T-BED ERRORS
FAME	REQUIREMENTS	118	36	45	98	79	25	41	77	36
FAME	DESIGN	57	21	22	52	67	21	36	65	30
IS/1	CODING	90	27	21	47	177	68	65	58	54
NODAL	TESTING	0	0	0	0	0	0	0	0	0
IS/1	DOC/MAINT	0	0	0	0	0	0	0	0	0
MAPPER	DOC/MAINT	22	8	4	10	7	5	4	10	0
OPTIMA	MANAGEMENT	0	0	0	0	0	0	0	0	0

DMAAC + DMAHTC - by tool

	TRAINING				DEVELOPMENT				
	L-HRS	C-HRS	T-RUNS	U-ERRORS	L-HRS	C-HRS	T-RUNS	U-ERRORS	T-ERRORS
FAME	220	61	74	157	225	65	96	159	83
SDDL	35	8	12	7	100	32	27	7	19
IS/1	90	27	21	47	177	68	65	58	54
TX	15	5	3	9	128	65	111	45	18
NODAL	4	0	0	8	17	3	0	0	0
MAPPER	30	10	9	10	12	11	16	14	0
OPTIMA	0	0	0	0	0	0	0	0	0
FORMAT	8	0	0	0	0	0	0	0	0

DMAAC + DMAHTC

	LABOR HOURS	COMP. HOURS	TOOL RUNS	USAGE ERRORS	T-BED ERRORS
FAME	445	126	170	316	83
SDDL	135	40	39	14	19
IS/1	267	95	86	105	54
TX	143	70	114	54	18
NODAL	21	3	0	8	0
MAPPER	42	21	25	24	0
OPTIMA	0	0	0	0	0
FORMAT	8	0	0	0	0
	----	---	---	---	---
	1061	355	434	521	174

```
.DATE 17 NOV 81  08:18:03  RID       4    29 OCT 81  THATCHER
.0820130            DATA COLLECTION MATRIX FOR : DISPLAY                  81462
* ------- TOOLS -------  .------- TOOL TRAINING ------------- .------ TEST-BED PROBLEM DEVELOPMENT --------- . ----.
* TOOL  .     LIFE       .LABOR .COMPUTER .  NR OF   .NR USAGE .LABOR .COMPUTER.  NR OF   .NR USAGE .NR TEST-   . RES-.
* USED  .   CYCLE PHASE  .HOURS . HOURS   .TOOL RUNS. ERRORS  .HOURS . HOURS   .TOOL RUNS. ERRORS   .BED ERRORS. ULT .
*=======.===============.======.=========.=========.=========.======.=========.=========.=========.==========.=====.
*XXXXXXX NAME OF TOOL BEING USED
*        XXXXXXXXXXXXXXXX LIFE CYCLE SUPPORTED BY TOOL
*                         XXXXXX TIME SPENT IN FORMAL PRESENTATIONS OR
*                                STUDYING MANUALS
*                                XXXXXXXXX CONNECT TIME USED IN ANALYZING
*                                          TOOL CAPABILITIES
*                                          XXXXXXXXX NUMBER OF TIMES TOOL
*                                                    ACCESSED FOR TRAINING
*                        SYNTAX ERRORS MADE WHILE    XXXXXXXXX
*                        USING THE TOOL DURING TRAINING OR ERRORS FOUND
*                        IN THE TOOL ITSELF
*                                                              XXXXXX TIME SPENT IN DEVELOPING PROBLEM
*                                                                     XXXXXXXX CONNECT TIME ASSOCIATED WITH TEST-
*                                                                              BED PROBLEM DEVELOPMENT
*                                                                              XXXXXXXXX TOOL ACCESSES FOR PROBLEM
*                                                                                        DEVELOPMENT
*                                                                     SYNTAX ERRORS MADE XXXXXXXXX
*                                                                     USING THE TOOL FOR PROBLEM DEVELOPMENT
*                                                                                ERRORS FOUND BY   XXXXXXXX
*                                                                                TOOL IN TEST-BED PROBLEM
*                                                                                                TEMPORARY  XXXXX
*                                                                                                RESULT FIELD

                     ..... END REPORT .....
```

.DATE 30 NOV 81 12:55:51 RID 8 04 NOV 81 THATCHER
.8820130 DATA COLLECTION MATRIX FOR : KINOSFATHER 81462

TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING LABOR HOURS	TOOL TRAINING COMPUTER HOURS	TOOL TRAINING NR OF TOOL RUNS	TOOL TRAINING NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT NR USAGE ERRORS	NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	4	0	0	0	4	0	0	0	0	
SDOL	DESIGN	0	0	0	0	0	0	0	0	0	
TX	CODING	0	0	0	0	0	0	0	0	0	
NODAL	TESTING	0	0	0	0	0	0	0	0	0	
MAPPER	DATA BASE	1	1	5	0	2	1	3	0	0	
OPTIMA	MANAGEMENT	0	0	0	0	0	0	0	0	0	
FORMAT	DOCUMENTATION	0	0	0	0	0	0	0	0	0	

..... END REPORT .....

.DATE 30 NOV 81 12:52:13 RID 3 04 NOV 81 THATCHER
. DATA COLLECTION MATRIX FOR : SPONS 81462

TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING LABOR HOURS	TOOL TRAINING COMPUTER HOURS	TOOL TRAINING NR OF TOOL RUNS	TOOL TRAINING NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT NR USAGE ERRORS	NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	15	1	4	4	35	5	6	2	3	
SDOL	DESIGN	3	1	2	2	25	1	2	1	2	
TX	CODING	1	1	1	1	9	1	1	1	1	
NODAL	TESTING										
MAPPER	DATA BASE										
OPTIMA	MANAGEMENT										
FORMAT	DOCUMENTATION										

..... END REPORT .....

.DATE 30 NOV 81 12:48:54 RID 4 04 NOV 81 THATCHER
. DATA COLLECTION MATRIX FOR : PINK 81462

TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING LABOR HOURS	TOOL TRAINING COMPUTER HOURS	TOOL TRAINING NR OF TOOL RUNS	TOOL TRAINING NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT NR USAGE ERRORS	NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	17	1	1	3	8	3	3	3	2	
SDOL	DESIGN	20	6	10	5	29	7	5	2	0	
TX	CODING	8	3	2	8	18	11	16	13	0	
NODAL	TESTING										
MAPPER	DATA BASE										
OPTIMA	MANAGEMENT										
FORMAT	DOCUMENTATION										

..... END REPORT .....

.DATE 20 NOV 81 09:47:36 RID 5 04 NOV 81 THATCHER
.8820130 DATA COLLECTION MATRIX FOR : RICE 81462

TOOLS: TOOL USED	TOOLS: LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	TOOL TRAINING: COMPUTER HOURS	TOOL TRAINING: NR OF TOOL RUNS	TOOL TRAINING: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT: COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT: NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	4	1	2	0	2	1	0	0	0	
SDDL	DESIGN	5	0			1	1				
TX	CODING	1	1			4	2				
NODAL	TESTING	3									
MAPPER	DATA BASE	5	1			1	1				
OPTIMA	MANAGEMENT										
FORMAT	DOCUMENTATION										

..... END REPORT .....

.DATE 20 NOV 81 13:09:47 RID 6 04 NOV 81 THATCHER
.8820130 DATA COLLECTION MATRIX FOR : HOLMGREN 81462

TOOLS: TOOL USED	TOOLS: LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	TOOL TRAINING: COMPUTER HOURS	TOOL TRAINING: NR OF TOOL RUNS	TOOL TRAINING: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT: COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT: NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	4	0	0	0	19	2	2	5	1	
SDDL	DESIGN	4	0	0	0	29	1	1	0	0	
TX	CODING	4	0	0	0	68	30	60	15	0	
NODAL	TESTING	0	0	0	0	12	0	0	0	0	
MAPPER	DATA BASE	1	0	0	0	1	1	2	0	0	
OPTIMA	MANAGEMENT	0	0	0	0	0	0	[illegible]	0	0	
FORMAT	DOCUMENTATION	8	0	0	0	0	0	[illegible]	0	0	

..... END REPORT .....

.DATE 20 NOV 81 09:01:57 RID 7 04 NOV 81 THATCHER
. DATA COLLECTION MATRIX FOR : MASBACK . 81462

TOOLS: TOOL USED	TOOLS: LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	TOOL TRAINING: COMPUTER HOURS	TOOL TRAINING: NR OF TOOL RUNS	TOOL TRAINING: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT: COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT: NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	01	01	00	00	11	08	08	06	11	
SDDL	DESIGN	01	01	00	00	16	22	19	04	17	
TX	CODING	02	01	00	00	29	21	34	16	09	
NODAL	TESTING	01	00	00	08	01	01	00	00	00	
MAPPER	DATA BASE	01	00	00	00	01	03	07	04	00	
OPTIMA	MANAGEMENT	00	00	00	00	00	00	00	00	00	
FORMAT	DOCUMENTATION	00	00	00	00	00	00	00	00	00	

..... END REPORT .....

.DATE 16 NOV 81 10:36:11 RID 11 10 NOV 81 THATCHER
.0820131 DATA COLLECTION MATRIX FOR MARTHA B1462

TOOLS: TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	COMPUTER HOURS	NR OF TOOL RUNS	NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	COMPUTER HOURS	NR OF TOOL RUNS	NR USAGE ERRORS	NR TEST-BED ERRORS	RESULT
FAME	REQUIREMENTS	10	1	1	2	8	4	5	4	5	0
FAME	DESIGN	14	1	2	5	15	5	3	4	6	0
IS/1	CODING	21	2	4	8	53	42	24	20	30	0
NODAL	TESTING	0	0	0	0	0	0	0	0	0	0
IS/1	DOC/MAINT	0	0	0	0	0	0	0	0	0	0
MAPPER	DATA BASE MGT	5	1	1	0	1	2	2	0	0	0
OPTIMA	PROJECT MGT	0	0	0	0	0	0	0	0	0	0

..... END REPORT .....

.DATE 16 NOV 81 12:10:36 RID 12 12 NOV 81 THATCHER
.0820130 DATA COLLECTION MATRIX FOR MARY B1462

TOOLS: TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	COMPUTER HOURS	NR OF TOOL RUNS	NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	COMPUTER HOURS	NR OF TOOL RUNS	NR USAGE ERRORS	NR TEST-BED ERRORS	RESULT
FAME	REQUIREMENTS	22	2	5	7	17	5	0	0	8	0
FAME	DESIGN	0	0	0	0	16	5	10	5	5	0
IS/1	CODING	5	4	0	0	32	5	10	0	5	0
NODAL	TESTING	0	0	0	0	0	0	0	0	0	0
IS/1	DOC/MAINT	0	0	0	0	0	0	0	0	0	0
MAPPER	DATA BASE MGT	0	0	0	0	0	0	0	0	0	0
OPTIMA	PROJECT MGT	0	0	0	0	0	0	0	0	0	0

..... END REPORT .....

.DATE 16 NOV 81 07:27:01 RID 14 10 NOV 81 THATCHER
.0820130 DATA COLLECTION MATRIX FOR GERI B1462

TOOLS: TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	COMPUTER HOURS	NR OF TOOL RUNS	NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	COMPUTER HOURS	NR OF TOOL RUNS	NR USAGE ERRORS	NR TEST-BED ERRORS	RESULT
FAME	REQUIREMENTS	26	15	10	30	11	5	5	25	3	0
FAME	DESIGN	6	2	2	8	6	3	3	15	3	0
IS/1	CODING	30	6	6	7	25	6	5	10	0	0
NODAL	TESTING	0	0	0	0	0	0	0	0	0	0
IS/1	DOC/MAINT	0	3	0	0	0	0	0	0	0	0
MAPPER	DATA BASE MGT	7	2	2	5	2	1	1	5	0	0
OPTIMA	PROJECT MGT	0	0	0	0	0	0	0	0	0	0

..... END REPORT .....

.DATE 16 NOV 81 07:41:37 RID 15 10 NOV 81 THATCHER
.0820131 DATA COLLECTION MATRIX FOR MIKE 81462

TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	TOOL TRAINING: COMPUTER HOURS	TOOL TRAINING: NR OF TOOL RUNS	TOOL TRAINING: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT: COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT: NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	24	3	16	14	18	2	10	8	5	0
FAME	DESIGN	9	1	6	4	8	1	6	4	3	0
IS/1	CODING	8	1	2	1	37	9	20	3	9	0
NODAL	TESTING	0	0	0	0	0	0	0	0	0	0
IS/1	DOC/MAINT	0	0	0	0	0	0	0	0	0	0
MAPPER	DATA BASE MGT	5	0	0	0	0	0	0	0	0	0
OPTIMA	PROJECT MGT	0	0	0	0	0	0	0	0	0	0

..... END REPORT .....

.DATE 12 NOV 81 09:53:32 RID 16 10 NOV 81 THATCHER
.0820130 DATA COLLECTION MATRIX FOR PETER 81462

TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	TOOL TRAINING: COMPUTER HOURS	TOOL TRAINING: NR OF TOOL RUNS	TOOL TRAINING: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT: COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT: NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	11	2	2	15	10	4	6	20	10	
FAME	DESIGN	5	2	2	20	7	4	4	20	10	
IS/1	CODING	2	4	5	25	22	4	3	20	10	
NODAL	TESTING	0	0	0	0	0	0				
IS/1	DOC/MAINT	0	0	0	0	0	0				
MAPPER	DATA BASE MGT	0	5	1	5	4	2	1	5	0	
OPTIMA	PROJECT MGT	0	0	0	0	0	0				

..... END REPORT .....

.DATE 12 NOV 81 09:29:18 RID 13 10 NOV 81 THATCHER
.0820130 DATA COLLECTION MATRIX FOR JOHNNIE 81462

TOOL USED	LIFE CYCLE PHASE	TOOL TRAINING: LABOR HOURS	TOOL TRAINING: COMPUTER HOURS	TOOL TRAINING: NR OF TOOL RUNS	TOOL TRAINING: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: LABOR HOURS	TEST-BED PROBLEM DEVELOPMENT: COMPUTER HOURS	TEST-BED PROBLEM DEVELOPMENT: NR OF TOOL RUNS	TEST-BED PROBLEM DEVELOPMENT: NR USAGE ERRORS	TEST-BED PROBLEM DEVELOPMENT: NR TEST-BED ERRORS	RES-ULT
FAME	REQUIREMENTS	25	13	11	30	15	5	15	20	5	
FAME	DESIGN	23	15	10	15	15	3	10	15	3	
IS/1	CODING	24	10	4	6	8	2	3	5	0	
NODAL	TESTING	0	0	0	0	0	0	0	0	0	
IS/1	DOC/MAINT	0	0	0	0	0	0	0	0	0	
MAPPER	DATA BASE MGT	5	0	0	0	0	0	0	0	0	
OPTIMA	PROJECT MGT	0	0	0	0	0	0	0	0	0	

..... END REPORT .....

APPENDIX G
CONCEPT IMPLEMENTATION EVALUATION MATRIX

NEEDS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
PROJECT MANAGEMENT																							
10 IMPROVED MILESTONE IDENTIFICATION							X		X														
12 IMPROVE MANLOADING							X	X	X			X											
14 IMPROVE SCHEDULE IMPACT ANALYSIS							X	X	X														
48 CHARGEBACK SYSTEM																					X		
56 MANAGEMENT TRACKING FUNCTIONS	X					X	X	X	X												X		
REQUIREMENTS																							
1 FORMAL REQUIREMENTS SPECIFICATION	X												X										
5 REQUIREMENTS TRACKING													X										
57 SOFTWARE DEVELOPMENT TOOLS	X	X									X		X					X		X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X			X							X			
DESIGN																							
21 SIMULATOR FOR DESIGN											X							X					
22 PROGRAM DESIGN LANGUAGE											X			X									
57 SOFTWARE DEVELOPMENT TOOLS	X	X									X			X				X		X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X				X						X			
CODING																							
55 MODERN SOURCE DATA ENTRY TECHNIQUES			X	X											X								
57 SOFTWARE DEVELOPMENT TOOLS	X	X													X			X		X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X					X					X		X	
TEST																							
2 QA PROCEDURES AND GUIDELINES										X										X			
21 SIMULATOR FOR DESIGN											X							X					
36 GRAPHICS AIDS	X			X														X					
57 SOFTWARE DEVELOPMENT TOOLS	X	X									X							X	X	X			
59 STANDARDIZED PHASED DEVELOPMENT	X									X									X	X			
MAINTENANCE																							
9 CONFIGURATION CONTROL	X		X		X																		
40 HISTORICAL DATA BASE TECHNIQUES							X										X						
57 SOFTWARE DEVELOPMENT TOOLS	X	X			X													X	X	X			
58 PRODUCTION PROGRAM OPTIMIZATION	X																		X			X	
59 STANDARDIZED PHASED DEVELOPMENT	X				X					X										X			
OTHER																							
3 INTERACTIVE SYSTEM ACCESS				X																			
4 INCREASED NUMBER OF TERMINALS				X																			
11 DECREASED PAPERWORK	X		X			X	X									X	X						
16 UPDATE OF OLD DOCUMENTATION	X																						
18 FASTER INTEGRATION OF NEW EMPLOYEES		X								X		X			X								X
34 AUTOMATED TEXT MANAGEMENT TOOL	X					X									X	X							
41 ORGANIZATION TOOLS/TECHNIQUES INTERFACE		X	X	X	X								X	X									
42 USER ASSISTANCE FUNCTION					X																		X
44 ERROR RATE STANDARDS																					X		X
46 REDUCE ACCOUNTING DATA REPORT ANOMALIES																	X				X		
47 COMPREHENSIVE TRAINING PROGRAM										X		X											
52 DECREASE TURNAROUND TIME TO MINUTES				X																	X		
54 NATURAL LANGUAGE USER/SYSTEM INTERFACE														X									
60 STANDARDIZED DEVELOPMENT HARDWARE			X	X																			

NEEDS

IMPLEMENTATION MATRIX

X : IMPLEMENTATION ID'D
X : NO IMPLEMENTATION

CONCEPTS

1. INTEGRATED SUPPORT DEVELOPMENT SYS
2. HIGH ORDER LANGUAGE
3. SINGLE LARGE MULTI-USER ENVIRONMENT
4. STANDARD SMALL MULTIPLE ENV'S
5. CONFIGURATION CONTROL SYSTEM
6. AUTOMATED OFFICE
7. PROJECT MANAGEMENT SYSTEM
8. COST ESTIMATING SYSTEMS
9. PROJECT PATH ANALYSIS METHOD
10. SOFTWARE ENGINEERING PRACTICES [illegible]
11. RAPID PROTOTYPING
12. AUTOMATED TRAINING PROGRAM
13. AUTOMATED REQUIREMENTS GENERATION
14. SOFTWARE DESIGN LANGUAGE
15. STRUCTURED PROGRAMMING FACILITY
16. INTERACTIVE TEXT PROCESSING
17. AUTOMATED DATA COLLECTION
18. INTERACTIVE SUPPORT SIMULATION SOFT
19. SOFTWARE TESTING SYSTEM
20. SOFTWARE STANDARDIZATION
21. CHARGEBACK SYSTEM
22. STRUCTURED PROGRAMMING
23. USER ASSISTANCE FUNCTION

SD/BSD Central Center

211

APPENDIX H
SUMMARY STATISTICS FROM EVALUATIONS

CIE_BY_NEED

FAME	#13 Auto.Req.Gen	# 1 For.Req.Spec	798.2
LARE	#13 Auto.Req.Gen	# 1 For.Req.Spec	772.2
PSL/PSA	#13 Auto.Req.Gen	# 1 For.Req.Spec	808.6
SRIMP	#13 Auto.Req.Gen	# 1 For.Req.Spec	595.4
			2974.4
CAVS	#20 Soft.Std	# 2 QA.Procs&Guides	723.6
FAVS	#20 Soft.Std	# 2 QA.Procs&Guides	1036.9
FTN 77 ANA	#20 Soft.Std	# 2 QA.Procs&Guides	720.0
			2480.4
HARRIS-SES	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	900.0
PDP 11/UNIX	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	840.0
SEL-SFTOOL80	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	870.0
VAX-IS/1	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	955.0
			3565.0
HARRIS	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	892.4
PDP 11/70	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	933.8
SEL	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	851.0
VAX	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	952.2
			3629.4
RDP 1100	#13 Auto.Req.Gen	# 5 Requirements Tracking	378.4
			378.4
FASP	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	672.0
IS/1 PWB	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	816.0
SOFTOOL II	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	644.0
SOLID	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	560.0
UNIX	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	756.0
CCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	860.0
SCCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	968.0
SLIB	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	744.0
SMS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	464.0
SOFTOOL II	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	628.0
SPMS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	824.0
			7936.0
OPTIMA	# 7 Proj.Mgt.Sys	#10 Imp.Milest.Id	663.0
CPAT	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	519.0
CPM	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	375.0
PERT	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	489.0
			2046.0
IS/1 INmail	# 1 Int.Spt.Dev.Sys	#11 Decr.Ppr'wk	432.0
UNIVAC-UNADS	# 3 Sg.Lg.Mult-Us.Env	#11 Decr.Ppr'wk	352.0
IS/1 INmail	# 6 Automated Off	#11 Decr.Ppr'wk	318.0
CPAT	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	410.0
OPTIMA	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	442.0
RDP 1100	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	322.0
SCERT II	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	388.0
IS/1 INword	#16 Int.Txt.Proc	#11 Decr.Ppr'wk	376.0
UTS4K PROC	#16 Int.Txt.Proc	#11 Decr.Ppr'wk	370.0
D'ly PlanIt	#17 Auto.Data.Coll	#11 Decr.Ppr'wk	466.0
			3876.0
OPTIMA	# 7 Proj.Mgt.Sys	#12 Improve Manloading	618.8
PRICE	# 7 Proj.Mgt.Sys	#12 Improve Manloading	546.0
SCERT II	# 7 Proj.Mgt.Sys	#12 Improve Manloading	543.2
SLIM	# 7 Proj.Mgt.Sys	#12 Improve Manloading	532.0

COCOMO	# 8 Cost.Est.Sys	#12 Improve Manloading	378.0
PRICE	# 8 Cost.Est.Sys	#12 Improve Manloading	532.0
SLIM	# 8 Cost.Est.Sys	#12 Improve Manloading	515.2
CPM	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	475.0
PERT	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	456.4
SCERT II	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	442.4
ASET	#12 Auto.Trng.Pgm	#12 Improve Manloading	366.8
			5405.8
CPAT	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	533.0
OPTIMA	# 7 Proj.Mgt.Sys.	#14 Impr.Schd.Impc.Ana	574.6
PRICE	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	507.0
SCERT II	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	504.4
SLIM	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	494.0
COCOMO	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	351.0
PRICE	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	494.0
SLIM	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	478.4
CPAT	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	449.8
CPM	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	325.0
PERT	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	423.8
			5135.0
FASP	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc	470.4
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc	599.2
			1069.6
DUAL	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's	411.4
FORTRAN 77	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's	572.0
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#18 Fstr.Int.New.Empl's	279.4
ASET	#12 Auto.Trng.Pgm	#18 Fstr.Int.New.Empl's	366.8
IPF	#15 SPF	#18 Fstr.Int.New.Empl's	272.8
FEDSIM	#23 User.Asst.Func	#18 Fstr.Int.New.Empl's	143.0
			2045.4
CS4	#11 Rapid Prototype	#21 Simulator for Design	355.2
PAWS	#11 Rapid Prototype	#21 Simulator for Design	310.4
USEIT	#11 Rapid Prototype	#21 Simulator for Design	400.0
			1065.6
CS4	#11 Rapid Prototyping	#22 PDL	799.2
SHELL	#11 Rapid Prototype	#22 PDL	835.2
USEIT	#11 Rapid Prototype	#22 PDL	900.0
PDL	#14 Soft.Dsgn.Lang	#22 PDL	921.6
SDDL	#14 Soft.Dsgn.Lang	#22 PDL	1119.6
			4575.6
IS/1 INword	# 1 Int.Spt.Dev.Sys	#34 Auto.Txt.Mgt.Sys	767.6
IS/1 INed	# 6 Automated Off	#34 Auto.Txt.Mgt.Sys	611.8
IPF	#15 SPF	#34 Auto.Txt.Mgt.Sys	471.2
IS/1 INword	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys	714.4
UTS4K PROC	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys	703.0
			3268.0
HARRIS	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids	561.0
SEL	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids	727.6
VAX	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids	727.6
			2016.2
MAPPER	# 7 Proj.Mgt.Sys	#40 Hist.DB.Tech's	233.2
D'ly PlanIt	#17 Auto.Data.Coll	#40 Hist.DB.Tech's	605.8
			839.0
ADA	# 2 High-Order Lang	#41 Org.Tools/Tech's.Int	452.2
FORTRAN 77	# 2 High-Order Lang	#41 Org.Tools/Tech's.Int	884.0

SOFTOOL II	# 5 Conf.Cntl.Sys	#41 Org.Tools/Tech's.Int	533.8
FAME	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1043.8
LARE	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1009.8
PSL/PSA	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1057.4
RDP 1100	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	584.8
SRIMP	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	778.6
PDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	870.4
SDDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	1057.4
			8272.2
CCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	774.0
SCCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	871.2
SLIB	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	669.6
SMS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	417.6
SOFTOOL II	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	565.2
SPMS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	741.6
FEDSIM	#23 User.Asst.Func	#42 User.Asst.Func	234.0
			4273.2
FEDSIM	#23 User.Asst.Func	#44 Error Rate Standards	169.0
			169.0
D'ly PlanIt	#17 Auto.Data.Coll	#46 Red.Acct.Data.Rept.Anom	652.4
			652.4
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#47 Comp.Trng.Pgm	482.6
ASET	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm	497.8
SOFTOOL 80	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm	425.6
			1406.0
PlanIt BlBk	#21 Chargeback System	#48 Chargeback System	465.8
			465.8
HARRIS	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	659.6
PDP 11/70	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	690.2
SEL	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	629.0
VAX	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	703.8
			703.8
UIFOLA	#14 Soft.Dsgn.Lang	#54 Natl.Lang.User/Sys.Int	267.4
			267.4
UNIVAC-4K'S	# 3 Sg.Lg.Mult-Us.Env	#55 Mod.Src.Data.Ent.Tech's	1021.2
IS/1	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	910.8
SES	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	837.2
UNIX	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	800.4
IPF	#15 SPF	#55 Mod.Src.Data.Ent.Tech's	570.4
			4140.0
MAPPER	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	902.4
OPTIMA	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	707.2
PRICE	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	624.0
RDP 1100	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	515.2
SCERT II	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	620.8
SLIM	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	608.0
COCOMO	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	432.0
PRICE	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	608.0
SLIM	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	588.8
CPAT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	553.6
CPM	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	400.0
PERT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	521.6
SCERT II	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	505.6
			7587.2
FASP	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	705.6

IS/1 PWB	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	886.2
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	898.8
SOLID	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	588.0
ADA	# 2 High-Order Lang	#57 Soft.Dev.Tools	558.6
DUAL	# 2 High-Order Lang	#57 Soft.Dev.Tools	785.4
FORTRAN 77	# 2 High-Order Lang	#57 Soft.Dev.Tools	1050.0
CS4	#11 Rapid Prototyping	#57 Soft.Dev.Tools	932.4
PAWS	#11 Rapid Prototyping	#57 Soft.Dev.Tools	814.8
USEIT	#11 Rapid Prototyping	#57 Soft.Dev.Tools	1050.0
FAME	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1289.4
LARE	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1247.4
PSL/PSA	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1306.2
RDP 1100	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	722.4
SRIMP	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	961.8
ADF	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	684.6
PDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1075.2
SDDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1306.2
IPF	#15 SPF	#57 Soft.Dev.Tools	520.8
FAVS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1289.4
FTN 77 ANA	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	898.8
SCMS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1134.0
SOFTOOL 80	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1121.4
CAVS	#20 Soft.Std	#57 Soft.Dev.Tools	844.2
FAVS	#20 Soft.Std	#57 Soft.Dev.Tools	1209.6
FTN 77 ANA	#20 Soft.Std	#57 Soft.Dev.Tools	840.0
			24721.0
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#58 Prod.Pgm.Opt	856.0
CAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	784.0
FAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1228.0
FTN 77 ANA	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	856.0
SCMS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1080.0
SOFTOOL 80	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1068.0
			5872.0
APSE	# 1 Int.Spt.Dev.Sys	#59 Std.Phsd.Dev	457.2
ADA	#22 Structured Pgm	#59 Std.Phsd.Dev	414.0
			871.2
UNIVAC 11/62	# 3 Sg.Lg.Mult-Us.Env	#60 Std.Dev.Hd'wr	843.6
HARRIS	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	737.2
SEL	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	703.0
VAX	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	786.6
			3070.4

CIE BY TOOL

ADA	# 2 High-Order Lang	#41 Org.Tools/Tech's.Int	452.2
ADA	# 2 High-Order Lang	#57 Soft.Dev.Tools	558.6
ADA	#22 Structured Pgm	#59 Std.Phsd.Dev	414.0
			1424.8
ADF	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	684.6
			684.6
APSE	# 1 Int.Spt.Dev.Sys	#59 Std.Phsd.Dev	457.2
			457.2
ASET	#12 Auto.Trng.Pgm	#12 Improve Manloading	366.8
ASET	#12 Auto.Trng.Pgm	#18 Fstr.Int.New.Empl's	366.8
ASET	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm	497.8
			1231.4
CAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	784.0
CAVS	#20 Soft.Std	# 2 QA.Procs&Guides	723.6
CAVS	#20 Soft.Std	#57 Soft.Dev.Tools	844.2
			2351.8
CCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	860.0
CCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	774.0
			1634.0
COCOMO	# 8 Cost.Est.Sys	#12 Improve Manloading	378.0
COCOMO	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	351.0
COCOMO	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	432.0
			1161.0
CPAT	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	410.0
CPAT	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	533.0
CPAT	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	519.0
CPAT	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	449.8
CPAT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	553.6
			2465.4
CPM	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	375.0
CPM	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	475.0
CPM	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	325.0
CPM	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	400.0
			1575.0
CS4	#11 Rapid Prototype	#21 Simulator for Design	355.2
CS4	#11 Rapid Prototyping	#22 PDL	799.2
CS4	#11 Rapid Prototyping	#57 Soft.Dev.Tools	932.4
			2086.8
D'ly PlanIt	#17 Auto.Data.Coll	#11 Decr.Ppr'wk	466.0
D'ly PlanIt	#17 Auto.Data.Coll	#40 Hist.DB.Tech's	605.8
D'ly PlanIt	#17 Auto.Data.Coll	#46 Red.Acct.Data.Rept.Anom	652.4
			1724.2
DUAL	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's	411.4
DUAL	# 2 High-Order Lang	#57 Soft.Dev.Tools	785.4
			1196.8
FAME	#13 Auto.Req.Gen	# 1 For.Req.Spec	798.2
FAME	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1043.8
FAME	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1289.4
			3131.4
FASP	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	672.0
FASP	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc	470.4
FASP	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	705.6
			1848.0

FAVS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1289.4
FAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1228.0
FAVS	#20 Soft.Std	# 2 QA.Procs&Guides	1036.8
FAVS	#20 Soft.Std	#57 Soft.Dev.Tools	1209.6
			4736.8
FEDSIM	#23 User.Asst.Func	#18 Fstr.Int.New.Empl's	143.0
FEDSIM	#23 User.Asst.Func	#42 User.Asst.Func	234.0
FEDSIM	#23 User.Asst.Func	#44 Error Rate Standards	169.0
			546.0
FORTRAN 77	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's	572.0
FORTRAN 77	# 2 High-Order Lang	#41 Org.Tools/Tech's.Int	884.0
FORTRAN 77	# 2 High-Order Lang	#57 Soft.Dev.Tools	1050.0
			2506.0
FTN 77 ANA	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	898.8
FTN 77 ANA	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	856.0
FTN 77 ANA	#20 Soft.Std	# 2 QA.Procs&Guides	720.0
FTN 77 ANA	#20 Soft.Std	#57 Soft.Dev.Tools	840.0
			3314.8
HARRIS	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	892.4
HARRIS	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids	561.0
HARRIS	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	659.6
HARRIS	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	737.2
HARRIS-SES	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	900.0
			3750.2
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#18 Fstr.Int.New.Empl's	279.4
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#47 Comp.Trng.Pgm	482.6
			762.0
IPF	#15 SPF	#18 Fstr.Int.New.Empl's	272.8
IPF	#15 SPF	#34 Auto.Txt.Mgt.Sys	471.2
IPF	#15 SPF	#55 Mod.Src.Data.Ent.Tech's	570.4
IPF	#15 SPF	#57 Soft.Dev.Tools	520.8
			1835.2
IS/1	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	910.8
			910.8
IS/1 INed	# 6 Automated Off	#34 Auto.Txt.Mgt.Sys	611.8
			611.8
IS/1 INmail	# 1 Int.Spt.Dev.Sys	#11 Decr.Ppr'wk	432.0
IS/1 INmail	# 6 Automated Off	#11 Decr.Ppr'wk	318.0
			750.0
IS/1 INword	# 1 Int.Spt.Dev.Sys	#34 Auto.Txt.Mgt.Sys	767.6
IS/1 INword	#16 Int.Txt.Proc	#11 Decr.Ppr'wk	376.0
IS/1 INword	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys	714.4
			1858.0
IS/1 PWB	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	816.0
IS/1 PWB	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	886.2
			1702.0
LARE	#13 Auto.Req.Gen	# 1 For.Req.Spec	772.2
LARE	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1009.8
LARE	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1247.4
			3029.4
MAPPER	# 7 Proj.Mgt.Sys	#40 Hist.DB.Tech's	233.2
MAPPER	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	902.4
			1135.6
OPTIMA	# 7 Proj.Mgt.Sys	#10 Imp.Milest.Id	663.0
OPTIMA	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	442.0

OPTIMA	# 7 Proj.Mgt.Sys	#12 Improve Manloading	618.8
OPTIMA	# 7 Proj.Mgt.Sys.	#14 Impr.Schd.Impc.Ana	574.6
OPTIMA	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	707.2
			3005.6
PlanIt BlBk	#21 Chargeback System	#48 Chargeback System	465.8
			465.8
PAWS	#11 Rapid Prototype	#21 Simulator for Design	310.4
PAWS	#11 Rapid Prototyping	#57 Soft.Dev.Tools	814.8
			1125.2
PDL	#14 Soft.Dsgn.Lang	#22 PDL	921.6
PDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	870.4
PDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1075.2
			2867.2
PDP 11/UNIX	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	840.0
			840.0
PDP 11/70	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	933.8
PDP 11/70	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	690.2
			1624.0
PERT	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	489.0
PERT	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	456.4
PERT	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	423.8
PERT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	521.6
			1890.8
PRICE	# 7 Proj.Mgt.Sys	#12 Improve Manloading	546.0
PRICE	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	507.0
PRICE	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	624.0
PRICE	# 8 Cost.Est.Sys	#12 Improve Manloading	532.0
PRICE	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	494.0
PRICE	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	608.0
			3311.0
PSL/PSA	#13 Auto.Req.Gen	# 1 For.Req.Spec	808.6
PSL/PSA	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1057.4
PSL/PSA	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1306.2
			3172.2
RDP 1100	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	322.0
RDP 1100	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	515.2
RDP 1100	#13 Auto.Req.Gen	# 5 Requirements Tracking	378.4
RDP 1100	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	584.8
RDP 1100	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	722.4
			2522.8
SCCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	968.0
SCCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	871.2
			1839.2
SCERT II	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	388.0
SCERT II	# 7 Proj.Mgt.Sys	#12 Improve Manloading	543.2
SCERT II	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	504.4
SCERT II	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	620.8
SCERT II	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	442.4
SCERT II	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	505.6
			3004.4
SCMS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1134.0
SCMS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1080.0
			2214.0
SDDL	#14 Soft.Dsgn.Lang	#22 PDL	1119.6
SDDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	1057.4

SDDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1306.2
			3483.2
SEL	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	851.0
SEL	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids	727.6
SEL	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	629.0
SEL	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	703.0
			2910.6
SEL-SFTOOL80	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	870.0
			870.0
SES	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	837.2
			837.2
SHELL	#11 Rapid Prototype	#22 PDL	835.2
			835.2
SLIB	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	744.0
SLIB	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	669.6
			1413.6
SLIM	# 7 Proj.Mgt.Sys	#12 Improve Manloading	532.0
SLIM	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	494.0
SLIM	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	608.0
SLIM	# 8 Cost.Est.Sys	#12 Improve Manloading	515.2
SLIM	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	478.4
SLIM	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	588.8
			3216.4
SMS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	464.0
SMS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	417.6
			881.6
SOFTOOL II	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	644.0
SOFTOOL II	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	628.0
SOFTOOL II	# 5 Conf.Cntl.Sys	#41 Org.Tools/Tech's.Int	533.8
SOFTOOL II	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	565.2
			2371.0
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc	599.2
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	898.8
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#58 Prod.Pgm.Opt	856.0
SOFTOOL 80	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm	425.6
SOFTOOL 80	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1121.4
SOFTOOL 80	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1068.0
			4969.0
SOLID	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	560.0
SOLID	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	588.0
			1148.0
SPMS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	824.0
SPMS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	741.6
			1565.6
SRIMP	#13 Auto.Req.Gen	# 1 For.Req.Spec	595.4
SRIMP	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	778.6
SRIMP	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	961.8
			2335.8
UIFOLA	#14 Soft.Dsgn.Lang	#54 Natl.Lang.User/Sys.Int	267.4
			267.4
UNIVAC 11/62	# 3 Sg.Lg.Mult-Us.Env	#60 Std.Dev.Hd'wr	843.6
			843.6
UNIVAC-UNADS	# 3 Sg.Lg.Mult-Us.Env	#11 Decr.Ppr'wk	352.0
			352.0
UNIVAC-4K'S	# 3 Sg.Lg.Mult-Us.Env	#55 Mod.Src.Data.Ent.Tech's	1021.2

			1021.2
UNIX	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	756.0
UNIX	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	800.4
			1556.4
USEIT	#11 Rapid Prototype	#21 Simulator for Design	400.0
USEIT	#11 Rapid Prototype	#22 PDL	900.0
USEIT	#11 Rapid Prototyping	#57 Soft.Dev.Tools	1050.0
			2350.0
UTS4K PROC	#16 Int.Txt.Proc	#11 Decr.Ppr'wk	370.0
UTS4K PROC	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys	703.0
			1073.0
VAX	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	952.2
VAX	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids	727.6
VAX	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time	703.8
VAX	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	786.6
			3170.2
VAX-IS/1	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	955.0
			955.0

CIE BY SCORE

PSL/PSA	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1306.2
SDDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1306.2
PAME	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1289.4
PAVS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1289.4
LARE	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1247.4
PAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1228.0
PAVS	#20 Soft.Std	#57 Soft.Dev.Tools	1209.6
SCMS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1134.0
SOFTOOL 80	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1121.4
SDDL	#14 Soft.Dsgn.Lang	#22 PDL	1119.6
SCMS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1080.0
PDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1075.2
SOFTOOL 80	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1068.0
PSL/PSA	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1057.4
SDDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	1057.4
FORTRAN 77	# 2 High-Order Lang	#57 Soft.Dev.Tools	1050.0
USEIT	#11 Rapid Prototyping	#57 Soft.Dev.Tools	1050.0
PAME	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1043.8
PAVS	#20 Soft.Std	# 2 QA.Procs&Guides	1036.8
UNIVAC-4K'S	# 3 Sg.Lg.Mult-Us.Env	#55 Mod.Src.Data.Ent.Tech's	1021.2
LARE	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1009.8
SCCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	968.0
SRIMP	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	961.8
VAX-IS/1	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	955.0
VAX	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	952.2
PDP 11/70	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	933.8
CS4	#11 Rapid Prototyping	#57 Soft.Dev.Tools	932.4
PDL	#14 Soft.Dsgn.Lang	#22 PDL	921.6
IS/1	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	910.8
MAPPER	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	902.4
HARRIS-SES	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	900.0
USEIT	#11 Rapid Prototype	#22 PDL	900.0
FTN 77 ANA	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	898.8
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	898.8
HARRIS	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	892.4
IS/1 PWB	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	886.2
FORTRAN 77	# 2 High-Order Lang	#41 Org.Tools/Tech's.Int	884.0
SCCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	871.2
SEL-SFTOOL80	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	870.0
PDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	870.4
CCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	860.0
FTN 77 ANA	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	856.0
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#58 Prod.Pgm.Opt	856.0
SEL	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's	851.0
CAVS	#20 Soft.Std	#57 Soft.Dev.Tools	844.2
UNIVAC 11/62	# 3 Sg.Lg.Mult-Us.Env	#60 Std.Dev.Hd'wr	843.6
PDP 11/UNIX	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	840.0
FTN 77 ANA	#20 Soft.Std	#57 Soft.Dev.Tools	840.0
SES	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	837.2
SHELL	#11 Rapid Prototype	#22 PDL	835.2
SPMS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	824.0
IS/1 PWB	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl	816.0
PAVS	#11 Rapid Prototyping	#57 Soft.Dev.Tools	814.8

PSL/PSA	#13 Auto.Req.Gen	# 1 For.Req.Spec		808.6
UNIX	# 4 Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's		800.4
CS4	#11 Rapid Prototyping	#22 PDL		799.2
FAME	#13 Auto.Req.Gen	# 1 For.Req.Spec		798.2
VAX	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr		786.6
DUAL	# 2 High-Order Lang	#57 Soft.Dev.Tools		785.4
CAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt		784.0
SRIMP	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int		778.6
CCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func		774.0
LARE	#13 Auto.Req.Gen	# 1 For.Req.Spec		772.2
IS/1 INword	# 1 Int.Spt.Dev.Sys	#34 Auto.Txt.Mgt.Sys		767.6
UNIX	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl		756.0
SLIB	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl		744.0
SPMS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func		741.6
HARRIS	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr		737.2
SEL	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids		727.6
VAX	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids		727.6
CAVS	#20 Soft.Std	# 2 QA.Procs&Guides		723.6
RDP 1100	#13 Auto.Req.Gen	#57 Soft.Dev.Tools		722.4
FTN 77 ANA	#20 Soft.Std	# 2 QA.Procs&Guides		720.0
IS/1 INword	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys		714.4
OPTIMA	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's		707.2
FASP	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools		705.6
UTS4K PROC	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys		703.0
VAX	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time		703.8
SEL	# 4 Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr		703.0
PDP 11/70	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time		690.2
ADF	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools		684.6
FASP	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl		672.0
SLIB	# 5 Conf.Cntl.Sys	#42 User.Asst.Func		669.6
OPTIMA	# 7 Proj.Mgt.Sys	#10 Imp.Milest.Id		663.0
HARRIS	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time		659.6
D'ly PlanIt	#17 Auto.Data.Coll	#46 Red.Acct.Data.Rept.Anom		652.4
SOFTOOL II	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl		644.0
SEL	# 4 Std.Sm.Mult.Env's	#52 Decr.Turn.Time		629.0
SOFTOOL II	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl		628.0
PRICE	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's		624.0
SCERT II	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's		620.8
OPTIMA	# 7 Proj.Mgt.Sys	#12 Improve Manloading		618.8
IS/1 INed	# 6 Automated Off	#34 Auto.Txt.Mgt.Sys		611.8
PRICE	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's		608.0
SLIM	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's		608.0
D'ly PlanIt	#17 Auto.Data.Coll	#40 Hist.DB.Tech's		605.8
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc		599.2
SRIMP	#13 Auto.Req.Gen	# 1 For.Req.Spec		595.4
SLIM	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's		588.8
SOLID	# 1 Int.Spt.Dev.Sys	#57 Soft.Dev.Tools		588.0
RDP 1100	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int		584.8
OPTIMA	# 7 Proj.Mgt.Sys.	#14 Impr.Schd.Impc.Ana		574.6
FORTRAN 77	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's		572.0
IPF	#15 SPF	#55 Mod.Src.Data.Ent.Tech's		570.4
SOFTOOL II	# 5 Conf.Cntl.Sys	#42 User.Asst.Func		565.2
HARRIS	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids		561.0
SOLID	# 1 Int.Spt.Dev.Sys	# 9 Conf.Cntl		560.0
ADA	# 2 High-Order Lang	#57 Soft.Dev.Tools		558.6

CPAT	# [illegible].Ana.Mth	#56 Mgt.Trkg.Func's	553.6
PRICE	# 7 [illegible]	#12 Improve Manloading	546.0
SCERT II	# 7 [illegible].Sys	#12 Improve Manloading	543.2
CPAT	# 7 [illegible].Sys	#14 Impr.Schd.Impc.Ana	533.0
SOFTOOL 11	# 5 [illegible].Sys	#41 Org.Tools/Tech's.Int	533.8
PRICE	# 6 [illegible] Sys	#12 Improve Manloading	532.0
SLIM	# [illegible].Mgt.Sys	#12 Improve Manloading	532.0
PERT	# 9 [illegible].Ana.Mth	#56 Mgt.Trkg.Func's	521.6
IPF	#15 [illegible]	#57 Soft.Dev.Tools	520.8
CPAT	# 9 [illegible].Ana.Mth	#10 Imp.Milest.Id	519.0
SLIM	# 8 [illegible].Sys	#12 Improve Manloading	515.2
RDP 1100	[illegible] Sys	#56 Mgt.Trkg.Func's	515.2
PRICE	[illegible] Sys	#14 Impr.Schd.Impc.Ana	507.0
SCERT II	[illegible].Mth	#56 Mgt.Trkg.Func's	505.6
SCERT II	# 7 [illegible].Sys	#14 Impr.Schd.Impc.Ana	504.4
ASET	[illegible].Pgm	#47 Comp.Trng.Pgm	497.8
PRICE	[illegible].Sys	#14 Impr.Schd.Impc.Ana	494.0
SLIM	[illegible] Sys	#14 Impr.Schd.Impc.Ana	494.0
PERT	[illegible].Ana.Mth	#10 Imp.Milest.Id	489.0
HYPERGR[illegible]	[illegible]	#47 Comp.Trng.Pgm	482.6
SLIM	[illegible].Sys	#14 Impr.Schd.Impc.Ana	478.4
CPM	[illegible] Mth	#12 Improve Manloading	475.0
IPF	[illegible]	#34 Auto.Txt.Mgt.Sys	471.2
FASP	[illegible].Dev.Sys	#16 Up.Old.Doc	470.4
D'ly Plan[illegible]	[illegible].Coll	#11 Decr.Ppr'wk	466.0
PlanIt [illegible]	[illegible]back System	#48 Chargeback System	465.8
SMS	[illegible].Sys	# 9 Conf.Cntl	464.0
APSE	[illegible].Dev.Sys	#59 Stg.Phsd.Dev	457.2
PERT	[illegible].Ana.Mth	#12 Improve Manloading	456.4
ADA	[illegible]-Order Lang	#41 Org.Tools/Tech's.Int	452.2
CPAT	[illegible].Ana.Mth	#14 Impr.Schd.Impc.Ana	449.8
OPTIMA	[illegible]	#11 Decr.Ppr'wk	442.0
SCERT II	[illegible].Ana.Mth	#12 Improve Manloading	442.4
IS/1 [illegible]	[illegible].Dev.Sys	#11 Decr.Ppr'wk	432.0
COCOMO	[illegible].Sys	#56 Mgt.Trkg.Func's	432.0
SOFT[illegible]	[illegible]	#47 Comp.Trng.Pgm	425.6
PERT	[illegible].Mth	#14 Impr.Schd.Impc.Ana	423.8
SMS	[illegible] Sys	#42 Heer.Asst.Func	417.6
ADA	[illegible] Lgn	#59 Stg.Phsd.Dev	414.0
DUAL	[illegible] Lang	#18 Fstr.Int.New.Empl's	411.4
CPAT	[illegible].Sys	#11 Decr.Ppr'wk	410.0
USEIT	# 1 [illegible] Prototype	#21 Simulator for Design	400.0
CPM	# 9 [illegible].Ana.Mth	#56 Mgt.Trkg.Func's	400.0
SCERT II	[illegible] Sys	#11 Decr.Ppr'wk	388.0
RDP 1100	[illegible]	# 5 Requirements Tracking	378.4
COCOMO	[illegible] Sys	#12 Improve Manloading	378.0
IS/1 [illegible]	[illegible].Proc	#11 Decr.Ppr'wk	376.0
CPM	[illegible].Ana.Mth	#10 Imp.Milest.Id	375.0
UTS4K [illegible]	[illegible].Proc	#11 Decr.Ppr'wk	370.0
ASET	[illegible] Pgm	#12 Improve Manloading	366.8
ASET	[illegible].Pgm	#18 Fstr.Int.New.Empl's	366.8
CS4	[illegible] Prototype	#21 Simulator for Design	355.2
UNIVAC-[illegible]	[illegible] Mult-Us.Env	#11 Decr.Ppr'wk	352.0
COCOMO	[illegible].Sys	#14 Impr.Schd.Impc.Ana	351.0
CPM	[illegible].Mth	#14 Impr.Schd.Impc.Ana	325.0

RDP 1100	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	322.0
IS/1 INmail	# 6 Automated Off	#11 Decr.Ppr'wk	318.0
PAWS	#11 Rapid Prototype	#21 Simulator for Design	310.4
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#18 Fstr.Int.New.Empl's	279.4
IPF	#15 SPF	#18 Fstr.Int.New.Empl's	272.8
UIFOLA	#14 Soft.Dsgn.Lang	#54 Natl.Lang.User/Sys.Int	267.4
FEDSIM	#23 User.Asst.Func	#42 User.Asst.Func	234.0
MAPPER	# 7 Proj.Mgt.Sys	#40 Hist.DB.Tech's	233.2
FEDSIM	#23 User.Asst.Func	#44 Error Rate Standards	169.0
FEDSIM	#23 User.Asst.Func	#18 Fstr.Int.New.Empl's	143.0

CIE BY CONCEPT

FASP	# 1	Int.Spt.Dev.Sys	# 9 Conf.Cntl	672.0
IS/1 PWB	# 1	Int.Spt.Dev.Sys	# 9 Conf.Cntl	816.0
SOFTOOL II	# 1	Int.Spt.Dev.Sys	# 9 Conf.Cntl	644.0
SOLID	# 1	Int.Spt.Dev.Sys	# 9 Conf.Cntl	560.0
UNIX	# 1	Int.Spt.Dev.Sys	# 9 Conf.Cntl	756.0
IS/1 INmail	# 1	Int.Spt.Dev.Sys	#11 Decr.Ppr'wk	432.0
FASP	# 1	Int.Spt.Dev.Sys	#16 Up.Old.Doc	470.4
SOFTOOL 80	# 1	Int.Spt.Dev.Sys	#16 Up.Old.Doc	599.2
IS/1 INword	# 1	Int.Spt.Dev.Sys	#34 Auto.Txt.Mgt.Sys	767.6
FASP	# 1	Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	705.6
IS/1 PWB	# 1	Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	886.2
SOFTOOL 80	# 1	Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	898.8
SOLID	# 1	Int.Spt.Dev.Sys	#57 Soft.Dev.Tools	588.0
SOFTOOL 80	# 1	Int.Spt.Dev.Sys	#58 Prod.Pgm.Opt	856.0
APSE	# 1	Int.Spt.Dev.Sys	#59 Std.Phsd.Dev	457.2
				10109.0
DUAL	# 2	High-Order Lang	#18 Fstr.Int.New.Empl's	411.4
FORTRAN 77	# 2	High-Order Lang	#18 Fstr.Int.New.Empl's	572.0
ADA	# 2	High-Order Lang	#41 Org.Tools/Tech's.Int	452.2
FORTRAN 77	# 2	High-Order Lang	#41 Org.Tools/Tech's.Int	884.0
ADA	# 2	High-Order Lang	#57 Soft.Dev.Tools	558.6
DUAL	# 2	High-Order Lang	#57 Soft.Dev.Tools	785.4
FORTRAN 77	# 2	High-Order Lang	#57 Soft.Dev.Tools	1050.0
				4713.6
UNIVAC-UNADS	# 3	Sg.Lg.Mult-Us.Env	#11 Decr.Ppr'wk	352.0
UNIVAC-4K'S	# 3	Sg.Lg.Mult-Us.Env	#55 Mod.Src.Data.Ent.Tech's	1021.2
UNIVAC 11/62	# 3	Sg.Lg.Mult-Us.Env	#60 Std.Dev.Hd'wr	843.6
				2216.8
HARRIS-SES	# 4	Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	900.0
PDP 11/UNIX	# 4	Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	840.0
SEL-SFTOOL80	# 4	Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	870.0
VAX-IS/1	# 4	Std.Sm.Mult.Env's	# 3 Int.Sys.Acc	955.0
HARRIS	# 4	Std.Sm.Mult.Env's	# 4 Incr.No.Term's	892.4
PDP 11/70	# 4	Std.Sm.Mult.Env's	# 4 Incr.No.Term's	933.8
SEL	# 4	Std.Sm.Mult.Env's	# 4 Incr.No.Term's	851.0
VAX	# 4	Std.Sm.Mult.Env's	# 4 Incr.No.Term's	952.2
HARRIS	# 4	Std.Sm.Mult.Env's	#36 Graphics Aids	561.0
SEL	# 4	Std.Sm.Mult.Env's	#36 Graphics Aids	727.6
VAX	# 4	Std.Sm.Mult.Env's	#36 Graphics Aids	727.6
HARRIS	# 4	Std.Sm.Mult.Env's	#52 Decr.Turn.Time	659.6
PDP 11/70	# 4	Std.Sm.Mult.Env's	#52 Decr.Turn.Time	690.2
SEL	# 4	Std.Sm.Mult.Env's	#52 Decr.Turn.Time	629.0
VAX	# 4	Std.Sm.Mult.Env's	#52 Decr.Turn.Time	703.8
IS/1	# 4	Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	910.8
SES	# 4	Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	837.2
UNIX	# 4	Std.Sm.Mult.Env's	#55 Mod.Src.Data.Ent.Tech's	800.4
HARRIS	# 4	Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	737.2
SEL	# 4	Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	703.0
VAX	# 4	Std.Sm.Mult.Env's	#60 Std.Dev.Hd'wr	786.6
				16668.4
CCS	# 5	Conf.Cntl.Sys	# 9 Conf.Cntl	860.0
SCCS	# 5	Conf.Cntl.Sys	# 9 Conf.Cntl	968.0
SLIB	# 5	Conf.Cntl.Sys	# 9 Conf.Cntl	744.0

SMS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	464.0
SOFTOOL II	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	628.0
SPMS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl	824.0
SOFTOOL II	# 5 Conf.Cntl.Sys	#41 Org.Tools/Tech's.Int	533.8
CCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	774.0
SCCS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	871.2
SLIB	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	669.6
SMS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	417.6
SOFTOOL II	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	565.2
SPMS	# 5 Conf.Cntl.Sys	#42 User.Asst.Func	741.6
			9061.0
IS/1 INmail	# 6 Automated Off	#11 Decr.Ppr'wk	318.0
IS/1 INed	# 6 Automated Off	#34 Auto.Txt.Mgt.Sys	611.8
			929.8
OPTIMA	# 7 Proj.Mgt.Sys	#10 Imp.Milest.Id	663.0
CPAT	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	410.0
OPTIMA	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	442.0
RDP 1100	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	322.0
SCERT II	# 7 Proj.Mgt.Sys	#11 Decr.Ppr'wk	388.0
OPTIMA	# 7 Proj.Mgt.Sys	#12 Improve Manloading	618.8
PRICE	# 7 Proj.Mgt.Sys	#12 Improve Manloading	546.0
SCERT II	# 7 Proj.Mgt.Sys	#12 Improve Manloading	543.2
SLIM	# 7 Proj.Mgt.Sys	#12 Improve Manloading	532.0
CPAT	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	533.0
OPTIMA	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	574.6
PRICE	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	507.0
SCERT II	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	504.4
SLIM	# 7 Proj.Mgt.Sys	#14 Impr.Schd.Impc.Ana	494.0
MAPPER	# 7 Proj.Mgt.Sys	#40 Hist.DB.Tech's	233.2
MAPPER	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	902.4
OPTIMA	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	707.2
PRICE	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	624.0
RDP 1100	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	515.2
SCERT II	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	620.8
SLIM	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's	608.0
			11288.4
COCOMO	# 8 Cost.Est.Sys	#12 Improve Manloading	378.0
PRICE	# 8 Cost.Est.Sys	#12 Improve Manloading	532.0
SLIM	# 8 Cost.Est.Sys	#12 Improve Manloading	515.2
COCOMO	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	351.0
PRICE	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	494.0
SLIM	# 8 Cost.Est.Sys	#14 Impr.Schd.Impc.Ana	478.4
COCOMO	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	432.0
PRICE	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	608.0
SLIM	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's	588.8
			4377.4
CPAT	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	519.0
CPM	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	375.0
PERT	# 9 Prj.Pth.Ana.Mth	#10 Imp.Milest.Id	489.0
CPM	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	475.0
PERT	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	456.4
SCERT II	# 9 Prj.Pth.Ana.Mth	#12 Improve Manloading	442.4
CPAT	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	449.8
CPM	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	325.0
PERT	# 9 Prj.Pth.Ana.Mth	#14 Impr.Schd.Impc.Ana	423.8

CPAT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	553.6
CPM	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	400.0
PERT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	521.6
SCERT II	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's	505.6
			5936.2
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#18 Fstr.Int.New.Empl's	279.4
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#47 Comp.Trng.Pgm	482.6
			762.0
CS4	#11 Rapid Prototype	#21 Simulator for Design	355.2
PAWS	#11 Rapid Prototype	#21 Simulator for Design	310.4
USEIT	#11 Rapid Prototype	#21 Simulator for Design	400.0
CS4	#11 Rapid Prototyping	#22 PDL	799.2
SHELL	#11 Rapid Prototype	#22 PDL	835.2
USEIT	#11 Rapid Prototype	#22 PDL	900.0
CS4	#11 Rapid Prototyping	#57 Soft.Dev.Tools	932.4
PAWS	#11 Rapid Prototyping	#57 Soft.Dev.Tools	814.8
USEIT	#11 Rapid Prototyping	#57 Soft.Dev.Tools	1050.0
			6397.2
ASET	#12 Auto.Trng.Pgm	#12 Improve Manloading	366.8
ASET	#12 Auto.Trng.Pgm	#18 Fstr.Int.New.Empl's	366.8
ASET	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm	497.8
SOFTOOL 80	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm	425.6
			1657.0
FAME	#13 Auto.Req.Gen	# 1 For.Req.Spec	798.2
LARE	#13 Auto.Req.Gen	# 1 For.Req.Spec	772.2
PSL/PSA	#13 Auto.Req.Gen	# 1 For.Req.Spec	808.6
SRIMP	#13 Auto.Req.Gen	# 1 For.Req.Spec	595.4
RDD 1100	#13 Auto.Req.Gen	# 5 Requirements Tracking	378.4
FAME	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1043.8
LARE	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1009.8
PSL/PSA	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	1057.4
RDD 1100	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	584.8
SRIMP	#13 Auto.Req.Gen	#41 Org.Tools/Tech's.Int	778.6
FAME	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1289.4
LARE	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1247.4
PSL/PSA	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	1306.2
RDD 1100	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	722.4
SRIMP	#13 Auto.Req.Gen	#57 Soft.Dev.Tools	961.8
			13354.4
PDL	#14 Soft.Dsgn.Lang	#22 PDL	921.6
SDDL	#14 Soft.Dsgn.Lang	#22 PDL	1119.6
PDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	870.4
SDDL	#14 Soft.Dsgn.Lang	#41 Org.Tools/Tech's.Int	1057.4
UIFOLA	#14 Soft.Dsgn.Lang	#54 Natl.Lang.User/Sys.Int	267.4
ADF	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	684.6
PDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1075.2
SDDL	#14 Soft.Dsgn.Lang	#57 Soft.Dev.Tools	1306.2
			7302.4
IPF	#15 SPF	#18 Fstr.Int.New.Empl's	272.8
IPF	#15 SPF	#34 Auto.Txt.Mgt.Sys	471.2
IPF	#15 SPF	#55 Mod.Src.Data.Ent.Tech's	570.4
IPF	#15 SPF	#57 Soft.Dev.Tools	520.8
			1835.2
IS/1 INvcl:	#16 Int.Txt.Proc	#11 Decr.Ppr'wk	376.0
UTSWF PROC	#16 Int.Txt.Proc	#11 Decr.Ppr'wk	370.0

IS/1 INword	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys	714.4
UTS4K PROC	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys	703.0
			2163.4
D'ly PlanIt	#17 Auto.Data.Coll	#11 Decr.Ppr'wk	466.0
D'ly PlanIt	#17 Auto.Data.Coll	#40 Hist.DB.Tech's	605.8
D'ly PlanIt	#17 Auto.Data.Coll	#46 Red.Acct.Data.Rept.Anom	652.4
			1724.2
FAVS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1289.4
FTN 77 ANA	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	898.8
SCMS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1134.0
SOFTOOL 80	#19 Soft.Test.Sys	#57 Soft.Dev.Tools	1121.4
CAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	784.0
FAVS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1228.0
FTN 77 ANA	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	856.0
SCMS	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1080.0
SOFTOOL 80	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt	1068.0
			9459.6
CAVS	#20 Soft.Std	# 2 QA.Procs&Guides	723.6
FAVS	#20 Soft.Std	# 2 QA.Procs&Guides	1036.8
FTN 77 ANA	#20 Soft.Std	# 2 QA.Procs&Guides	720.0
CAVS	#20 Soft.Std	#57 Soft.Dev.Tools	844.2
FAVS	#20 Soft.Std	#57 Soft.Dev.Tools	1209.6
FTN 77 ANA	#20 Soft.Std	#57 Soft.Dev.Tools	840.0
			5374.2
PlanIt BlBk	#21 Chargeback System	#48 Chargeback System	465.8
			465.8
ADA	#22 Structured Pgm	#59 Std.Phsd.Dev	414.0
			414.0
FEDSIM	#23 User.Asst.Func	#18 Fstr.Int.New.Empl's	143.0
FEDSIM	#23 User.Asst.Func	#42 User.Asst.Func	234.0
FEDSIM	#23 User.Asst.Func	#44 Error Rate Standards	169.0
			546.0

BEST_CASE_BY_NEED

ADA	#22 Structured Pgm	#59 Std.Phsd.Dev
APSE	# 1 Int.Spt.Dev.Sys	#59 Std.Phsd.Dev
ASET	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm
CPAT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's
D'ly PlanIt	#17 Auto.Data.Coll	#11 Decr.Ppr'wk
		#40 Hist.DB.Tech's
		#46 Red.Acct.Data.Rept.Anom
FAVS	#20 Soft.Std	# 2 QA.Procs&Guides
	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt
FEDSIM	#23 User.Asst.Func	#44 Error Rate Standards
FORTRAN 77	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#47 Comp.Trng.Pgm
IPF	#15 SPF	#55 Mod.Src.Data.Ent.Tech's
IS/1 INed	# 6 Automated Off	#34 Auto.Txt.Mgt.Sys
IS/1 INword	# 1 Int.Spt.Dev.Sys	#34 Auto.Txt.Mgt.Sys
	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys
MAPPER	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's
OPTIMA	# 7 Proj.Mgt.Sys	#10 Imp.Milest.Id
		#12 Improve Manloading
		#14 Impr.Schd.Impc.Ana
PlanIt BlBk	#21 Chargeback System	#48 Chargeback System
PRICE	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's
PSL/PSA	#13 Auto.Req.Gen	# 1 For.Req.Spec
		#41 Org.Tools/Tech's.Int
		#57 Soft.Dev.Tools
RDP 1100	#13 Auto.Req.Gen	# 5 Requirements Tracking
SCCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl
		#42 User.Asst.Func
SDDL	#14 Soft.Dsgn.Lang	#22 PDL
		#41 Org.Tools/Tech's.Int
		#57 Soft.Dev.Tools
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc
UIFOLA	#14 Soft.Dsgn.Lang	#54 Natl.Lang.User/Sys.Int
UNIVAC 11/62	# 3 Sg.Lg.Mult-Us.Env	#60 Std.Dev.Hd'wr
UNIVAC-4K'S	# 3 Sg.Lg.Mult-Us.Env	#55 Mod.Src.Data.Ent.Tech's
USEIT	#11 Rapid Prototype	#21 Simulator for Design
VAX	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's
		#36 Graphics Aids
		#52 Decr.Turn.Time
VAX-IS/1	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc

BEST_CASE_BY_CONCEPT

ADA	#22 Structured Pgm	#59 Std.Phsd.Dev
ASET	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm
CPAT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's
D'ly PlanIt	#17 Auto.Data.Coll	#11 Decr.Ppr'wk
		#40 Hist.DB.Tech's
		#46 Red.Acct.Data.Rept.Anom
FAVS	#19 Soft.Test.Sys	#57 Soft.Dev.Tools
		#58 Prod.Pgm.Opt
	#20 Soft.Std	# 2 QA.Procs&Guides
		#57 Soft.Dev.Tools
FEDSIM	#23 User.Asst.Func	#42 User.Asst.Func
		#44 Error Rate Standards
FORTRAN 77	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's
		#57 Soft.Dev.Tools
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#47 Comp.Trng.Pgm
IPF	#15 SPF	#55 Mod.Src.Data.Ent.Tech's
IS/1 INed	# 6 Automated Off	#34 Auto.Txt.Mgt.Sys
IS/1 INword	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys
MAPPER	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's
OPTIMA	# 7 Proj.Mgt.Sys	#10 Imp.Milest.Id
		#12 Improve Manloading
		#14 Impr.Schd.Impc.Ana
PlanIt BlBk	#21 Chargeback System	#48 Chargeback System
PRICE	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's
PSL/PSA	#13 Auto.Req.Gen	# 1 For.Req.Spec
		#41 Org.Tools/Tech's.Int
		#57 Soft.Dev.Tools
RDP 1100	#13 Auto.Req.Gen	# 5 Requirements Tracking
SCCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl
SDDL	#14 Soft.Dsgn.Lang	#22 PDL
		#57 Soft.Dev.Tools
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc
		#57 Soft.Dev.Tools
UIFOLA	#14 Soft.Dsgn.Lang	#54 Natl.Lang.User/Sys.Int
UNIVAC 11/62	# 3 Sg.Lg.Mult-Us.Env	#60 Std.Dev.Hd'wr
UNIVAC-4K'S	# 3 Sg.Lg.Mult-Us.Env	#55 Mod.Src.Data.Ent.Tech's
USEIT	#11 Rapid Prototype	#21 Simulator for Design
		#57 Soft.Dev.Tools
VAX	# 4 Std.Sm.Mult.Env's	#36 Graphics Aids
		#52 Decr.Turn.Time
VAX-IS/1	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc

BEST_CASE_MODERN_PROGRAMMING_ENVIRONMENT_FOR_DMA

ADA	#22 Structured Pgm	#59 Std.Phsd.Dev
APSE	# 1 Int.Spt.Dev.Sys	#59 Std.Phsd.Dev
ASET	#12 Auto.Trng.Pgm	#47 Comp.Trng.Pgm
CPAT	# 9 Prj.Pth.Ana.Mth	#56 Mgt.Trkg.Func's
D'ly PlanIt	#17 Auto.Data.Coll	#11 Decr.Ppr'wk
		#40 Hist.DB.Tech's
		#46 Red.Acct.Data.Rept.Anom
FAVS	#20 Soft.Std	# 2 QA.Procs&Guides
		#57 Soft.Dev.Tools
	#19 Soft.Test.Sys	#58 Prod.Pgm.Opt
		#57 Soft.Dev.Tools
FEDSIM	#23 User.Asst.Func	#44 Error Rate Standards
		#42 User.Asst.Func
FORTRAN 77	# 2 High-Order Lang	#18 Fstr.Int.New.Empl's
		#57 Soft.Dev.Tools
HYPERGRPH	#10 Sft.Eng.Prt.Trg	#47 Comp.Trng.Pgm
IPF	#15 SPF	#55 Mod.Src.Data.Ent.Tech's
IS/1 INed	# 6 Automated Off	#34 Auto.Txt.Mgt.Sys
IS/1 INword	# 1 Int.Spt.Dev.Sys	#34 Auto.Txt.Mgt.Sys
	#16 Int.Txt.Proc	#34 Auto.Txt.Mgt.Sys
MAPPER	# 7 Proj.Mgt.Sys	#56 Mgt.Trkg.Func's
OPTIMA	# 7 Proj.Mgt.Sys	#10 Imp.Milest.Id
		#12 Improve Manloading
		#14 Impr.Schd.Impc.Ana
PlanIt BlBk	#21 Chargeback System	#48 Chargeback System
PRICE	# 8 Cost.Est.Sys	#56 Mgt.Trkg.Func's
PSL/PSA	#13 Auto.Req.Gen	# 1 For.Req.Spec
		#41 Org.Tools/Tech's.Int
		#57 Soft.Dev.Tools
RDP 1100	#13 Auto.Req.Gen	# 5 Requirements Tracking
SCCS	# 5 Conf.Cntl.Sys	# 9 Conf.Cntl
		#42 User.Asst.Func
SDDL	#14 Soft.Dsgn.Lang	#22 PDL
		#41 Org.Tools/Tech's.Int
		#57 Soft.Dev.Tools
SOFTOOL 80	# 1 Int.Spt.Dev.Sys	#16 Up.Old.Doc
		#57 Soft.Dev.Tools
UIFOLA	#14 Soft.Dsgn.Lang	#54 Natl.Lang.User/Sys.Int
UNIVAC 11/62	# 3 Sg.Lg.Mult-Us.Env	#60 Std.Dev.Hd'wr
UNIVAC-4K'S	# 3 Sg.Lg.Mult-Us.Env	#55 Mod.Src.Data.Ent.Tech's
USEIT	#11 Rapid Prototype	#21 Simulator for Design
		#57 Soft.Dev.Tools
VAX	# 4 Std.Sm.Mult.Env's	# 4 Incr.No.Term's
		#36 Graphics Aids
		#52 Decr.Turn.Time
VAX-IS/1	# 4 Std.Sm.Mult.Env's	# 3 Int.Sys.Acc

APPENDIX I
LORAN NAVIGATIONAL LATTICE PROBLEM FOR USE.IT EVALUATION

original: 04 August 1982
revised: 18 August 1982
(indicated by marginal vertical lines)

DEMONSTRATION USE.IT PROBLEM FOR DEFENSE MAPPING AGENCY STUDY (Contract no. F30602-81-C-0039) LORAN NAVIGATIONAL LATTICE PROBLEM

1.0 INTRODUCTION

The Navigational Lattice Problem is a typical DMA mapping problem which will be used to demonstrate the USE.IT tool capabilities. The solution to the problem is a set or lattice of hyperbolas each indicating a different constant distance from a master-slave pair of radio transmitting stations. The LORAN operator tunes his receiver to a master-slave pair of stations and reads a time delay in microseconds. This time delay represents the time difference in receiving the radio signal from the master station and from the slave station. The constant distance is directly proportional to the time delay; the proportionality constant being the radio propagation velocity. The formula used to determine the hyperbolas is

V= Time B + Time C + Time D - Time A, where V is a constant.

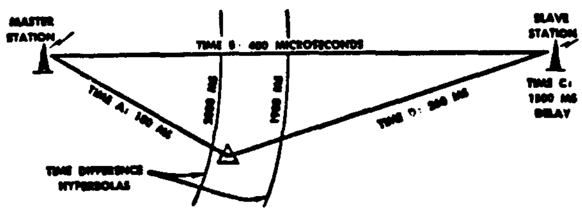


FIGURE 1

As an example, Figure 1 shows that the ship or aircraft is determined to be on the 1980 μsec hyperbola because

$$V = (400 + 1500 + 260 - 180)\mu sec = 1980\ \mu sec$$

2.0 SCOPE OF PROBLEM FOR USE.IT DEMONSTRATION

The purpose of the LORAN problem is to demonstrate the capabilities of the USE.IT tool. Hence, we shall assume a simplified mathematical model for the LORAN line calculation, and emphasize the cartographic aspects of the problem. The following assumptions simplify the mathematical model:

1. Assume a flat earth.
2. Disregard time delay complexitites caused by radio propagation over water or ground and reflections from the E and F layers of the ionosphere.
3. Consider only one master-slave pair of stations.

We will employ the USE.IT tool to produce a LORAN chart as shown in Figure 2.

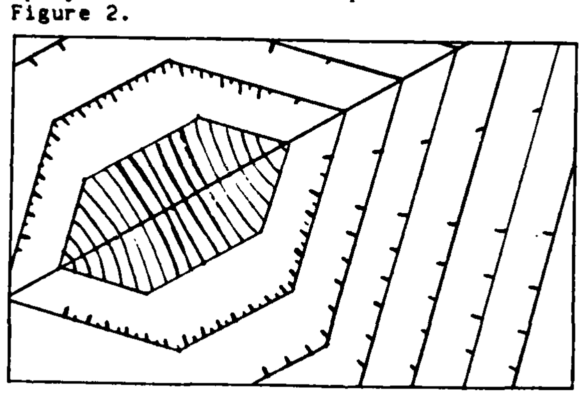


FIGURE 2

Origin

The major characteristics of this chart will be:

1. Two LORAN stations will be specified.
2. Smooth, complete hyperbolic curves will be drawn within the interior hexagon.
3. The points of intersection of the LORAN lines with the exterior hexagons will be shown.
4. The hexagons will be uniformly spaced and concentric out to the chart boundary.
5. A windowing capability will be provided to simulate the placement of the LORAN data on a paper chart.
6. The baseline between the master station and the slave station will be drawn and extended to the chart boundaries if either station or both stations are within the chart boundary.
7. The chart boundary will be a rectangle and it will be drawn.

The most significant input data are planned to be:

1. Position Definition Parameters
   a. Master Station Location input as (latitude, longitude)
   b. Slave Station Location input as (latitude, longitude)
2. Cartographic Option Parameters
   a. Chart Window Boundary Points input as (latitude, longitude)
   b. Hyperbolic Curve Spacing in chart dimensions
   c. Hexagon Spacing
   d. Overall Scaling

We plan to demonstrate the following two scenarios:

1. On the HOS VAX system, employ USE.IT for requirements definition and Fortran source code generation, and employ the VAX support software to execute the code on the VAX with output to a VT100 graphics terminal.
2. On the UNIVAC system, demonstrate the compatibility of USE.IT produced Fortran source code with UNIVAC support software by creating a UNIVAC produced magnetic tape of the LORAN data that is subsequently plotted off-line.

The Fortran source produced by the RAT function of USE.IT will be transported to a UNIVAC system where it will be compiled and executed. In a like manner the Fortran source can be compiled and executed on the VAX system. The outputs differ in that the VAX output will be a VT100 terminal and the UNIVAC output will be a magnetic tape, for subsequent off-line plotting. The scenarios are shown in Figure 3.

3.0 HIGH-LEVEL TREE STRUCTURE FOR THE LORAN PROBLEM

Figure 4 shows the top-down, high-level tree structure for the LORAN problem. This figure shows the requirements for the problem--not its implementation.

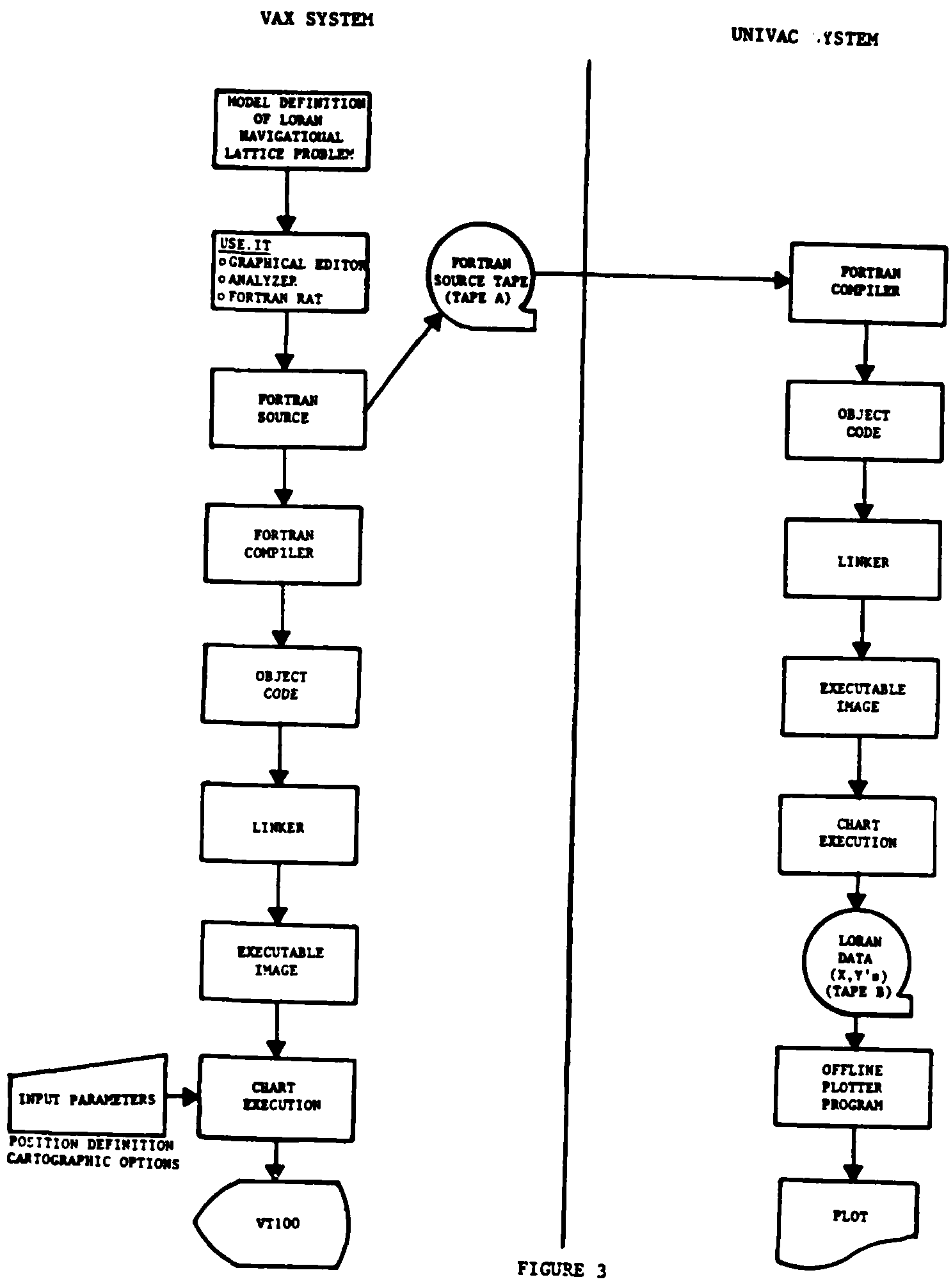

VAX SYSTEM
UNIVAC SYSTEM
MODEL DEFINITION OF LORAN NAVIGATIONAL LATTICE PROBLEM
USE.IT
o GRAPHICAL EDITOR
o ANALYZER
o FORTRAN RAT
FORTRAN SOURCE
FORTRAN SOURCE TAPE (TAPE A)
FORTRAN COMPILER
OBJECT CODE
LINKER
EXECUTABLE IMAGE
INPUT PARAMETERS
POSITION DEFINITION
CARTOGRAPHIC OPTIONS
CHART EXECUTION
VT100
FORTRAN COMPILER
OBJECT CODE
LINKER
EXECUTABLE IMAGE
CHART EXECUTION
LORAN DATA (X,Y's) (TAPE B)
OFFLINE PLOTTER PROGRAM
PLOT

FIGURE 3

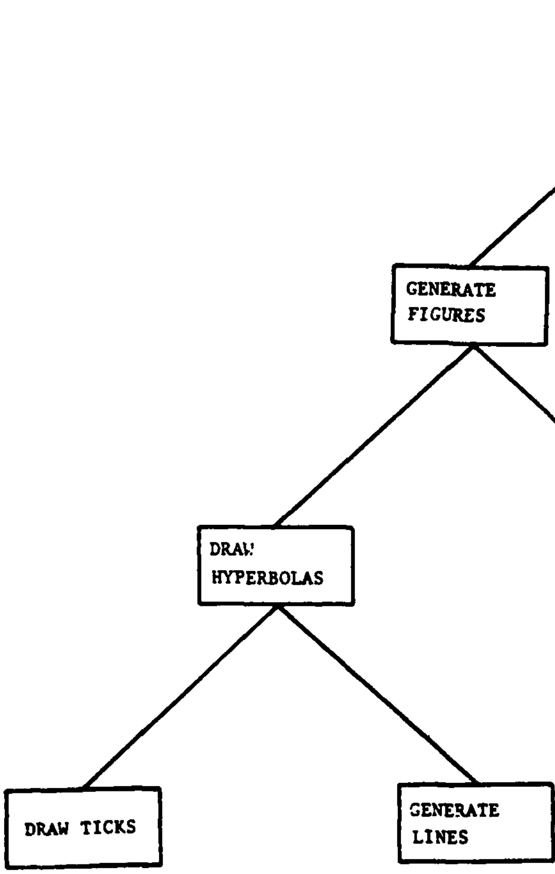

CREATE CHART
GENERATE FIGURES
GENERATE BORDER
DRAW HYPERBOLAS
DRAW HEXAGON
DRAW TICKS
GENERATE LINES

FIGURE 4

4.0 LIST OF CANDIDATE HIGH-LEVEL PRIMITIVES

Within the context of the LORAN problem, we consider a high-level primitive to have the characteristics that (1) it performs a generalized function in the requirements definition of a problem, and (2) its function may be useful to a larger class of DMA problems. These high-level primitives could form the basis of a library of functions that would be tailored tc the DMA environment and make requirements definition eãsier for the user. The following is a candidate list of high-level primitives that would be developed for the LORAN problem:

1. DRAW - draws a straight line segment between two specified points
2. TRANSLATE - translates a line by a specified displacement vector
3. ROTATE - rotates a line in two dimensions through a specified angle
4. SCALE - changes the dimensions of a line by a constant amount
5. WINDOW - selects a specified two dimensional region of a larger picture for display
6. PROJECTION - transforms (latitude, longitude) to (abscissa, ordinate)

We anticipate that these high-level primitives would be implemented by more fundamental primitives that would also be available in the library.

5.0 OUTPUT FORMAT

We will format the LORAN data for two output devices: (1) a VT100 graphics terminal and (2) a magnetic tape that can be read by the UNIVAC system. (See Figure 3 for the two demonstration scenarios.)

The output format to the VT100 graphics terminal will be compatible with the VT100 graphics terminal and the HOS VAX system. Since there is no interface with the UNIVAC system for this scenario, we anticipate no problems.

The UNIVAC scenario requires the specification of two output formats: (1) the magnetic tape format and (2) the LORAN data format. The magnetic tape format for the Fortran source code will be:

9 tracks
1600 bits per inch density
unlabelled
unblocked
80 column card images
ASCII format

Reference Tape A
in Figure 3

The LORAN data will be temporarily stored in an array A(2,201) as follows:

A(1,1) = the # of data points
A(2,1) = the feature index

and for $2 \leq I \leq 201$

Reference Tape B in Figure 3

$A(1,I) = X_{I-1}$
$A(2,I) = Y_{I-1}$

as demonstrated below.

	1	2	3		200	201
1	# of points	X_1	X_2	. . .	X_{199}	X_{200}
2	feature index	Y_1	Y_2	. . .	Y_{199}	Y_{200}

Once the array is full it will be either displayed on the screen or sent to a tape unit using an unformatted Fortran WRITE statement (see tape B of Figure 3) depending on whether the program is to be run on the VAX or Univac.

6.0 CANDIDATE FOR LORAN PROBLEM ENHANCEMENT

To demonstrate the capabilities of the USE.IT tool for software maintenance, we will enhance the capabilities of the initial LORAN problem solution by adding intermediate tick marks on the exterior hexagons of the LORAN chart. We will incorporate this enhancement at the requirements level, and we will add another high-level primitive TICK. TICK will place tick marks on a specified line segment at a specified interval. The intermediate tick marks will be a different length from the hyperbola intersection tick marks.

APPENDIX J
CONCEPT IMPLEMENTATION EVALUATION SHEETS

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1 PWB

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	5	3	15
System Resources			
Allocations Required	6	3	18

TOTAL	204.0
NEED WEIGHT	x 4.0
CIE SCORE	816.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: UNIX

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	4	2	8
Maturity	9	3	27
Vendor Support	5	1	5
Availability	8	3	24
Hardware Compatibility	10	3	30
Environment Compatibility	3	3	9
Government Access	3	1	3
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	3	2	6
Training	3	3	9
System Resources			
Allocations Required	6	3	18

TOTAL	189.0
NEED WEIGHT	x 4.0
CIE SCORE	756.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FASP

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Maturity	10	3	30
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	2	3	6
Environment Compatibility	2	3	6
Government Access	10	1	10
Flexibility of Use			
Hardware	3	2	6
Software	3	2	6
Conceptual Simplicity			
Use	7	2	14
Training	5	3	15
System Resources			
Allocations Required	4	3	12

TOTAL	168.0
NEED WEIGHT	x 4.0
CIE SCORE	672.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOFTOOL II

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	3	2	6
Maturity	1	3	3
Vendor Support	8	1	8
Availability	6	3	18
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
System Resources			
Allocations Required	0	3	0
TOTAL			161.0
NEED WEIGHT			x 4.0
CIE SCORE			644.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOLID

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	9	3	27
Support Documentation	0	2	0
Maturity	8	3	24
Vendor Support	1	1	1
Availability	1	3	3
Hardware Compatibility	7	3	21
Environment Compatibility	5	3	15
Government Access	1	1	1
Flexibility of Use			
Hardware	8	2	16
Software	8	2	16
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
System Resources			
Allocations Required	1	3	3

TOTAL	140.0
NEED WEIGHT	x 4.0
CIE SCORE	560.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1 INmail

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	1	2	2
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
System Resources			
Allocations Required	7	3	21

TOTAL	216.0
NEED WEIGHT	x 2.0
CIE SCORE	432.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOFTOOL 80

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #16 Update of Old Documentation

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Maturity	10	3	30
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
System Resources			
Allocations Required	0	3	0
TOTAL			214.0
NEED WEIGHT			x 2.8
CIE SCORE			599.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FASP

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #16 Update of Old Documentation

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Maturity	10	3	30
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	2	3	6
Environment Compatibility	2	3	6
Government Access	10	1	10
Flexibility of Use			
Hardware	3	2	6
Software	3	2	6
Conceptual Simplicity			
Use	7	2	14
Training	5	3	15
System Resources			
Allocations Required	4	3	12

TOTAL	168.0
NEED WEIGHT	x 2.8
CIE SCORE	470.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1 INword

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #34 Automated Text Management System

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	1	2	2
Conceptual Simplicity			
Use	8	2	16
Training	7	3	21
System Resources			
Allocations Required	7	3	21
TOTAL			202.0
NEED WEIGHT			x 3.8
CIE SCORE			767.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOFTOOL 80

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Maturity	10	3	30
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
System Resources			
Allocations Required	0	3	0

TOTAL	214.0
NEED WEIGHT	x 4.2
CIE SCORE	898.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FASP

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Maturity	10	3	30
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	2	3	6
Environment Compatibility	2	3	6
Government Access	10	1	10
Flexibility of Use			
Hardware	3	2	6
Software	3	2	6
Conceptual Simplicity			
Use	7	2	14
Training	5	3	15
System Resources			
Allocations Required	4	3	12
TOTAL			168.0
NEED WEIGHT			x 4.2
CIE SCORE			705.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOLID

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	9	3	27
Support Documentation	0	2	0
Maturity	8	3	24
Vendor Support	1	1	1
Availability	1	3	3
Hardware Compatibility	7	3	21
Environment Compatibility	5	3	15
Government Access	1	1	1
Flexibility of Use			
Hardware	8	2	16
Software	8	2	16
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
System Resources			
Allocations Required	1	3	3

TOTAL 140.0
NEED WEIGHT x 4.2
CIE SCORE 588.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1 PWB

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	6	2	12
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
System Resources			
Allocations Required	6	3	18

TOTAL	211.0
NEED WEIGHT	x4.2
CIE SCORE	886.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: ADA PROGRAMMING SUPPORT ENVIRONMENT

o FOR CONCEPT: #1 Integrated Support Dev. Sys.

o SATISFIES NEED: #59 Standardized Phased Development

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Maturity	1	3	3
Vendor Support	9	1	9
Availability	2	3	6
Hardware Compatibility	3	3	9
Environment Compatibility	3	3	9
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Software	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	2	3	6
System Resources			
Allocations Required	3	3	9

TOTAL	127.0
NEED WEIGHT	x 3.6
CIE SCORE	457.2

PRECEDING PAGE BLANK-NOT FILMED

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FORTRAN 77

o FOR CONCEPT: #2 High-Order Language

o SATISFIES NEED: #18 Faster Intergation of New Empl's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	10	2	20
Diagnostics			
Documentation	8	2	16
Interactive Support	6	2	12
Maturity	5	3	15
Availability	10	3	30
Hardware Compatibility	9	3	27
Environment Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	8	2	16
Software	9	2	18
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
System Resources			
Allocations Required	7	3	21

TOTAL	260.0
NEED WEIGHT	x 2.2
CIE SCORE	572.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: DUAL

o FOR CONCEPT: #2 High-Order Language

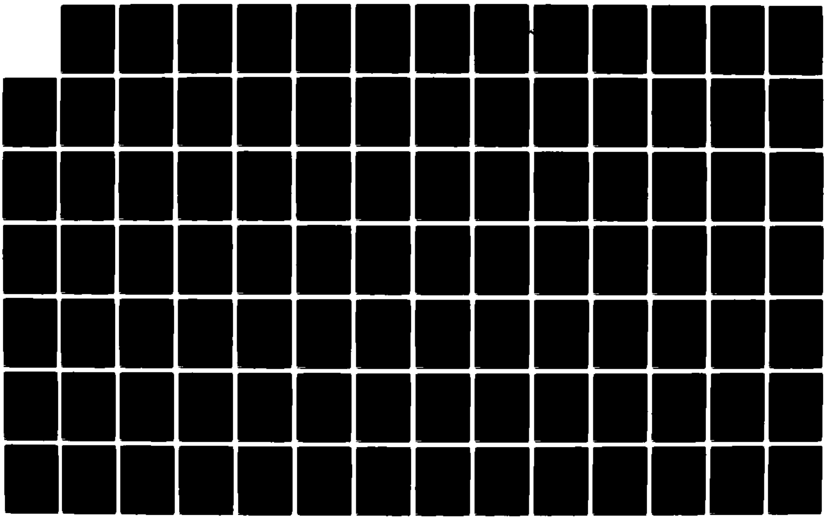
o SATISFIES NEED: #18 Faster Intergation of New Empl's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	9	3	27
Government Access	5	1	5
Flexibility of Use			
Hardware	10	2	20
Software	10	2	20
Conceptual Simplicity			
Use	5	2	10
Training	0	3	0
System Resources			
Allocations Required	5	3	15

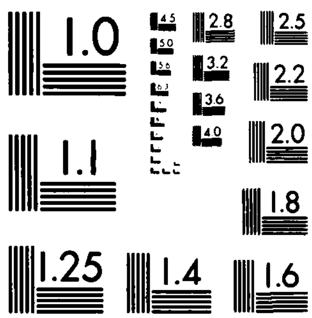
TOTAL	187.0
NEED WEIGHT	x 2.2
CIE SCORE	411.4

AD-A125 626 INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY VOLUME 1(U) GENERAL DYNAMICS FORT WORTH TX FORT WORTH DIV H C CONN ET AL. JAN 83 DMA-2-014-VOL-1 4/5

UNCLASSIFIED RADC-TR-83-3-VOL-1 F30602-81-C-0039 F/G 9/2 NL



M-2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FORTRAN 77

o FOR CONCEPT: #2 High-Order Language

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	10	2	20
Diagnostics			
Documentation	8	2	16
Interactive Support	6	2	12
Maturity	5	3	15
Availability	10	3	30
Hardware Compatibility	9	3	27
Environment Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	8	2	16
Software	9	2	18
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
System Resources			
Allocations Required	7	3	21

TOTAL	260.0
NEED WEIGHT	x 3.4
CIE SCORE	884.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: ADA

o FOR CONCEPT: #2 High-Order Language

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	10	2	20
Diagnostics			
Documentation	10	2	20
Interactive Support	8	2	16
Maturity	1	3	3
Availability	2	3	6
Hardware Compatibility	3	3	9
Environment Compatibility	3	3	9
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Software	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	2	3	6
System Resources			
Allocations Required	6	3	18

TOTAL	133.0
NEED WEIGHT	x 3.4
CIE SCORE	452.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FORTRAN 77

o FOR CONCEPT: #2 High-Order Language

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	8	2	16
Diagnostics			
Documentation	6	2	12
Interactive Support	4	2	8
Maturity	5	3	15
Availability	10	3	30
Hardware Compatibility	9	3	27
Environment Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	8	2	16
Software	9	2	18
Conceptual Simplicity			
Use	10	2	20
Training	9	3	27
System Resources			
Allocations Required	7	3	21

TOTAL	250.0
NEED WEIGHT	x 4.2
CIE SCORE	1050.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: ADA
- o FOR CONCEPT: #2 High-Order Language
- o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	10	2	20
Diagnostics			
Documentation	10	2	20
Interactive Support	8	2	16
Maturity	1	3	3
Availability	2	3	6
Hardware Compatibility	3	3	9
Environment Compatibility	3	3	9
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Software	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	2	3	6
System Resources			
Allocations Required	6	3	18

TOTAL	133.0
NEED WEIGHT	x 4.2
CIE SCORE	558.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: DUAL

o FOR CONCEPT: #2 High-Order Language

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	9	3	27
Government Access	5	1	5
Flexibility of Use			
Hardware	10	2	20
Software	10	2	20
Conceptual Simplicity			
Use	5	2	10
Training	0	3	0
System Resources			
Allocations Required	5	3	15

TOTAL	187.0
NEED WEIGHT	x 4.2
CIE SCORE	785.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: UNIVAC 1100/62 - UNADS

o FOR CONCEPT: #3 Single Large Multi-User Env.

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	7	3	21
Flexibility of Use			
Software	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	0	3	0
System Resources			
Capabilities Supported	5	3	15

TOTAL	176.0
NEED WEIGHT	x 2.0
CIE SCORE	352.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: UNIVAC 1100/62 - 4000 TERMINALS
- o FOR CONCEPT: #3 Single Large Multi-User Env.
- o SATISFIES NEED: #55 Modern Source Data Entry Tech's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	6	3	18
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	10	3	30
Flexibility of Use			
Software	7	2	14
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	7	3	21

TOTAL	222.0
NEED WEIGHT	x 4.6
CIE SCORE	1021.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: UNIVAC 1100/62

o FOR CONCEPT: #3 Single Large Multi-User Env.

o SATISFIES NEED: #60 Standardized Development Hardware

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	6	3	18
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	10	3	30
Flexibility of Use			
Software	7	2	14
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	7	3	21

TOTAL	222.0
NEED WEIGHT	x 3.8
CIE SCORE	843.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: VAX - IS/1

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #3 Interactive System Access

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	4	3	12
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	8	3	24

TOTAL	191.0
NEED WEIGHT	x 5.0
CIE SCORE	955.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SEL - SOFTOOL 80

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #3 Interactive System Access

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	10	3	30
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	1	3	3
Environment Compatibility	4	3	12
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	6	2	12
Training	6	3	18
System Resources			
Capabilities Supported	7	3	21

TOTAL	174.0
NEED WEIGHT	x 5.0
CIE SCORE	870.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: HARRIS - SES

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #3 Interactive System Access

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	8	3	24
Vendor Support	10	1	10
Availability	10	3	30
Hardware Compatibility	1	3	3
Environment Compatibility	5	3	15
Flexibility of Use			
Software	6	2	12
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	7	3	21

TOTAL	180.0
NEED WEIGHT	x 5.0
CIE SCORE	900.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PDP 11/70 - UNIX

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #3 Interactive System Access

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	9	3	27
Vendor Support	5	1	5
Availability	8	3	24
Hardware Compatibility	10	3	30
Environment Compatibility	3	3	9
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	3	2	6
Training	3	3	9
System Resources			
Capabilities Supported	6	3	18

TOTAL	168.0
NEED WEIGHT	x 5.0
CIE SCORE	840.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: VAX

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #4 Increased Number of Terminals

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	9	1	9
Availability	9	3	27
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Flexibility of Use			
Software	8	2	16
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	7	3	21

TOTAL	207.0
NEED WEIGHT	x 4.6
CIE SCORE	952.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SEL

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #4 Increased Number of Terminals

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	8	1	8
Availability	9	3	27
Hardware Compatibility	6	3	18
Environment Compatibility	6	3	18
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	5	3	15

TOTAL	185.0
NEED WEIGHT	x 4.6
CIE SCORE	851.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: HARRIS

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #4 Increased Number of Terminals

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	9	3	27
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	6	3	18
TOTAL			194.0
NEED WEIGHT			x 4.6
CIE SCORE			892.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PDP 11/70

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #4 Increased Number of Terminals

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	9	1	9
Availability	9	3	27
Hardware Compatibility	8	3	24
Environment Compatibility	9	3	27
Flexibility of Use			
Software	6	2	12
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	5	3	15

TOTAL	203.0
NEED WEIGHT	x 4.6
CIE SCORE	933.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SEL

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #36 Graphics Aids

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	9	3	27
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	7	3	21
Flexibility of Use			
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	7	3	21
System Resources			
Capabilities Supported	8	3	24

TOTAL	214.0
NEED WEIGHT	x 3.4
CIE SCORE	727.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: VAX

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #36 Graphics Aids

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	9	3	27
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	7	3	21
Flexibility of Use			
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	7	3	21
System Resources			
Capabilities Supported	8	3	24

TOTAL 214.0
NEED WEIGHT x 3.4
CIE SCORE 727.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: HARRIS

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #36 Graphics Aids

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Maturity	8	3	24
Vendor Support	10	1	10
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	1	3	3
Flexibility of Use			
Software	9	2	18
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	3	3	9

TOTAL	165.0
NEED WEIGHT	x 3.4
CIE SCORE	561.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: VAX

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #52 Decrease Turnaround Time

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	9	1	9
Availability	9	3	27
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Flexibility of Use			
Software	8	2	16
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	7	3	21

TOTAL	207.0
NEED WEIGHT	x 3.4
CIE SCORE	703.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: HARRIS

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #52 Decrease Turnaround Time

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	9	3	27
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	6	3	18

TOTAL	194.0
NEED WEIGHT	x 3.4
CIE SCORE	659.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o *IMPLEMENTATION: PDP 11/70*

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #52 Decrease Turnaround Time

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	9	1	9
Availability	9	3	27
Hardware Compatibility	8	3	24
Environment Compatibility	9	3	27
Flexibility of Use			
Software	6	2	12
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	5	3	15

TOTAL	203.0
NEED WEIGHT	x 3.4
CIE SCORE	690.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: SEL
- o FOR CONCEPT: #4 Standard Small Multiple Env's
- o SATISFIES NEED: #52 Decrease Turnaround Time

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	8	1	8
Availability	9	3	27
Hardware Compatibility	6	3	18
Environment Compatibility	6	3	18
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	5	3	15

TOTAL	185.0
NEED WEIGHT	x 3.4
CIE SCORE	629.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #55 Modern Source Data Entry Tech's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	6	3	18
Flexibility of Use			
Software	7	2	14
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	7	3	21

TOTAL	198.0
NEED WEIGHT	x 4.6
CIE SCORE	910.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: UNIX

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #55 Modern Source Data Entry Tech's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	10	3	30
Vendor Support	5	1	5
Availability	10	3	30
Hardware Compatibility	8	3	24
Environment Compatibility	3	3	9
Flexibility of Use			
Software	10	2	20
Conceptual Simplicity			
Use	1	2	2
Training	1	3	3
System Resources			
Capabilities Supported	7	3	21

TOTAL	174.0
NEED WEIGHT	x 4.6
CIE SCORE	800.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SES

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #55 Modern Source Data Entry Tech's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Maturity	8	3	24
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	2	3	6
Environment Compatibility	5	3	15
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	6	3	18

TOTAL	182.0
NEED WEIGHT	x 4.6
CIE SCORE	837.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: VAX

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #60 Standardized Development Hardware

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	9	1	9
Availability	9	3	27
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Flexibility of Use			
Software	8	2	16
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	7	3	21

TOTAL	207.0
NEED WEIGHT	x 3.8
CIE SCORE	786.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SEL

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #60 Standardized Development Hardware

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Maturity	10	3	30
Vendor Support	8	1	8
Availability	9	3	27
Hardware Compatibility	6	3	18
Environment Compatibility	6	3	18
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
System Resources			
Capabilities Supported	5	3	15

TOTAL	185.0
NEED WEIGHT	x 3.8
CIE SCORE	703.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: HARRIS

o FOR CONCEPT: #4 Standard Small Multiple Env's

o SATISFIES NEED: #60 Standardized Development Hardware

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	9	3	27
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Flexibility of Use			
Software	5	2	10
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Capabilities Supported	6	3	18
TOTAL			194.0
NEED WEIGHT			x 3.8
CIE SCORE			737.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCCS

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Diagnostics			
Documentation	3	2	6
Interactive Support	5	2	10
Automated Procedure	10	2	20
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	7	2	14
Conceptual Simplicity			
Use	7	2	14
Training	9	3	27
System Resources			
Allocations Required	5	3	15

TOTAL	242.0
NEED WEIGHT	x 4.0
CIE SCORE	968.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CCS

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	7	2	14
Diagnostics			
Documentation	3	2	6
Interactive Support	5	2	10
Automated Procedure	10	2	20
Maturity	10	3	30
Availability	5	3	15
Hardware Compatibility	3	3	9
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	7	2	14
Conceptual Simplicity			
Use	7	2	14
Training	9	3	27
System Resources			
Allocations Required	5	3	15

TOTAL	215.0
NEED WEIGHT	x 4.0
CIE SCORE	860.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIB

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	7	3	21
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	8	1	8
Flexibility of Use			
Hardware	7	2	14
Software	10	2	20
Conceptual Simplicity			
Use	5	2	10
Training	0	3	0
System Resources			
Allocations Required	5	3	15

TOTAL	186.0
NEED WEIGHT	x 4.0
CIE SCORE	744.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SMS

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	1	3	3
Availability	1	3	3
Hardware Compatibility	1	3	3
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	3	3	9
System Resources			
Allocations Required	0	3	0
TOTAL			116.0
NEED WEIGHT			x 4.0
CIE SCORE			464.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SPMS

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #9 Configuration Control

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Diagnostics			
Documentation	5	2	10
Interactive Support	5	2	10
Automated Procedure	5	2	10
Maturity	3	3	9
Availability	10	3	30
Hardware Compatibility	1	3	3
Environment Compatibility	1	3	3
Government Access	5	1	5
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Allocations Required	8	3	24

TOTAL	206.0
NEED WEIGHT	x 4.0
CIE SCORE	824.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o **IMPLEMENTATION: SOFTOOL II**

o **FOR CONCEPT: #5 Configuration Control System**

o **SATISFIES NEED: #9 Configuration Control**

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	3	2	6
Diagnostics			
Documentation	2	2	4
Interactive Support	0	2	0
Automated Procedure	0	2	0
Maturity	1	3	3
Availability	6	3	18
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
System Resources			
Allocations Required	0	3	0

TOTAL	157.0
NEED WEIGHT	x 4.0
CIE SCORE	628.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOFTOOL II

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	3	2	6
Diagnostics			
Documentation	2	2	4
Interactive Support	0	2	0
Automated Procedure	0	2	0
Maturity	1	3	3
Availability	6	3	18
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
System Resources			
Allocations Required	0	3	0

TOTAL 157.0
NEED WEIGHT x 3.4
CIE SCORE 533.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCCS

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #42 User Assistance Function

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Diagnostics			
Documentation	3	2	6
Interactive Support	5	2	10
Automated Procedure	10	2	20
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	7	2	14
Conceptual Simplicity			
Use	7	2	14
Training	9	3	27
System Resources			
Allocations Required	5	3	15

TOTAL	242.0
NEED WEIGHT	x 3.6
CIE SCORE	871.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: CCS
- o FOR CONCEPT: #5 Configuration Control System
- o SATISFIES NEED: #42 User Assistance Function

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	7	2	14
Diagnostics			
Documentation	3	2	6
Interactive Support	5	2	10
Automated Procedure	10	2	20
Maturity	10	3	30
Availability	5	3	15
Hardware Compatibility	3	3	9
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	7	2	14
Conceptual Simplicity			
Use	7	2	14
Training	9	3	27
System Resources			
Allocations Required	5	3	15

TOTAL	215.0
NEED WEIGHT	x 3.6
CIE SCORE	774.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIB

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #42 User Assistance Function

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	7	3	21
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	8	1	8
Flexibility of Use			
Hardware	7	2	14
Software	10	2	20
Conceptual Simplicity			
Use	5	2	10
Training	0	3	0
System Resources			
Allocations Required	5	3	15

TOTAL 186.0
NEED WEIGHT x 3.6
CIE SCORE 669.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SMS

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #42 User Assistance Function

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	1	3	3
Availability	1	3	3
Hardware Compatibility	1	3	3
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	3	3	9
System Resources			
Allocations Required	0	3	0
TOTAL			116.0
NEED WEIGHT			x 3.6
CIE SCORE			417.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SPMS

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #42 User Assistance Function

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Diagnostics			
Documentation	5	2	10
Interactive Support	5	2	10
Automated Procedure	5	2	10
Maturity	3	3	9
Availability	10	3	30
Hardware Compatibility	1	3	3
Environment Compatibility	1	3	3
Government Access	5	1	5
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
System Resources			
Allocations Required	8	3	24
TOTAL			206.0
NEED WEIGHT			x 3.6
CIE SCORE			741.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOFTOOL II

o FOR CONCEPT: #5 Configuration Control System

o SATISFIES NEED: #42 User Assistance Function

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	3	2	6
Diagnostics			
Documentation	2	2	4
Interactive Support	0	2	0
Automated Procedure	0	2	0
Maturity	1	3	3
Availability	6	3	18
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
System Resources			
Allocations Required	0	3	0

TOTAL	157.0
NEED WEIGHT	x 3.6
CIE SCORE	565.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1 INmail

o FOR CONCEPT: #6 Automated Office

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Automated Procedure	10	2	20
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	1	2	2
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27

TOTAL	159.0
NEED WEIGHT	x 2.0
CIE SCORE	318.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1 INed

o FOR CONCEPT: #6 Automated Office

o SATISFIES NEED: #34 Automated Text Management System

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Automated Procedure	9	2	18
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	7	3	21
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	8	2	16
Conceptual Simplicity			
Use	7	2	14
Training	4	3	12

TOTAL	161.0
NEED WEIGHT	x 3.8
CIE SCORE	611.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: OPTIMA

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #10 Improved Milestone Identification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	4	2	8
Training	4	3	12
Output			
DMA Applicable	5	3	15
Understandable	5	2	10
System Resources			
Allocations Required	5	3	15

TOTAL	221.0
NEED WEIGHT	x 3.0
CIE SCORE	663.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: OPTIMA

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	4	2	8
Training	4	3	12
Output			
DMA Applicable	5	3	15
Understandable	5	2	10
System Resources			
Allocations Required	5	3	15

TOTAL	221.0
NEED WEIGHT	x 2.0
CIE SCORE	442.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: RDP 1100

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	4	2	8
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	0	2	0
Training	5	3	15
Output			
DMA Applicable	5	3	15
Understandable	0	2	0
System Resources			
Allocations Required	5	3	15

TOTAL 161.0
NEED WEIGHT x 2.0
CIE SCORE 322.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CPAT

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16
System Resources			
Allocations Required	5	3	15
TOTAL			205.0
NEED WEIGHT			x 2.0
CIE SCORE			410.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCERT II

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Diagnostics			
Documentation	1	2	2
Interactive Support	1	2	2
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	9	3	27
Government Access	5	1	5
Flexibility of Use			
Hardware	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	0	3	0
Output			
DMA Applicable	5	3	15
Understandable	9	2	18
System Resources			
Allocations Required	0	3	0

TOTAL	194.0
NEED WEIGHT	x 2.0
CIE SCORE	388.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: OPTIMA

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	4	2	8
Training	4	3	12
Output			
DMA Applicable	5	3	15
Understandable	5	2	10
System Resources			
Allocations Required	5	3	15

TOTAL	221.0
NEED WEIGHT	x 2.8
CIE SCORE	618.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o *IMPLEMENTATION: PRICE*

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	1	2	2
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	5	3	15
Output			
DMA Applicable	4	3	12
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0

TOTAL	195.0
NEED WEIGHT	x 2.8
CIE SCORE	546.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIM

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	2	2	4
Interactive Support	7	2	14
Automated Procedure	7	2	14
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	0	3	0
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	4	3	12
Output			
DMA Applicable	4	3	12
Understandable	6	2	12
System Resources			
Allocations Required	0	3	0

TOTAL	190.0
NEED WEIGHT	x 2.8
CIE SCORE	532.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCERT II

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Diagnostics			
Documentation	1	2	2
Interactive Support	1	2	2
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	9	3	27
Government Access	5	1	5
Flexibility of Use			
Hardware	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	0	3	0
Output			
DMA Applicable	5	3	15
Understandable	9	2	18
System Resources			
Allocations Required	0	3	0

TOTAL	194.0
NEED WEIGHT	x 2.8
CIE SCORE	543.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: OPTIMA

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	4	2	8
Training	4	3	12
Output			
DMA Applicable	5	3	15
Understandable	5	2	10
System Resources			
Allocations Required	5	3	15
TOTAL			221.0
NEED WEIGHT			x 2.6
CIE SCORE			574.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PRICE

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	1	2	2
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	5	3	15
Output			
DMA Applicable	4	3	12
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0

TOTAL	195.0
NEED WEIGHT	x 2.6
CIE SCORE	507.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIM

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	2	2	4
Interactive Support	7	2	14
Automated Procedure	7	2	14
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	0	3	0
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	4	3	12
Output			
DMA Applicable	4	3	12
Understandable	6	2	12
System Resources			
Allocations Required	0	3	0

TOTAL	190.0
NEED WEIGHT	x 2.6
CIE SCORE	494.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCERT II

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Diagnostics			
Documentation	1	2	2
Interactive Support	1	2	2
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	9	3	27
Government Access	5	1	5
Flexibility of Use			
Hardware	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	0	3	0
Output			
DMA Applicable	5	3	15
Understandable	9	2	18
System Resources			
Allocations Required	0	3	0

TOTAL	194.0
NEED WEIGHT	x 2.6
CIE SCORE	504.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CPAT

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16
System Resources			
Allocations Required	5	3	15

TOTAL	205.0
NEED WEIGHT	x 2.6
CIE SCORE	533.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: MAPPER

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #40 Historical Data Base Techniques

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	8	2	16
Interactive Support	8	2	16
Automated Procedure	8	2	16
Maturity	8	3	24
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	9	2	18
Training	8	3	24
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	1	3	3

TOTAL	282.0
NEED WEIGHT	x 2.6
CIE SCORE	233.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: OPTIMA

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	7	2	14
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	4	2	8
Training	4	3	12
Output			
DMA Applicable	5	3	15
Understandable	5	2	10
System Resources			
Allocations Required	5	3	15

TOTAL	221.0
NEED WEIGHT	x 3.2
CIE SCORE	707.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: RDP 1100

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	4	2	8
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	0	2	0
Training	5	3	15
Output			
DMA Applicable	5	3	15
Understandable	0	2	0
System Resources			
Allocations Required	5	3	15
TOTAL			161.0
NEED WEIGHT			x 3.2
CIE SCORE			515.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: MAPPER

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	8	2	16
Interactive Support	8	2	16
Automated Procedure	8	2	16
Maturity	8	3	24
Availability	10	3	30
Hardware Compatibility	10	3	30
Governmant Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Conceptual Simplicity			
Use	9	2	18
Training	8	3	24
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	1	3	3

TOTAL	282.0
NEED WEIGHT	x 3.2
CIE SCORE	902.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PRICE

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	1	2	2
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	5	3	15
Output			
DMA Applicable	4	3	12
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0

TOTAL	195.0
NEED WEIGHT	x 3.2
CIE SCORE	624.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIM

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Diagnostics			
Documentation	2	2	4
Interactive Support	7	2	14
Automated Procedure	7	2	14
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	0	3	0
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	4	3	12
Output			
DMA Applicable	4	3	12
Understandable	6	2	12
System Resources			
Allocations Required	0	3	0

TOTAL	190.0
NEED WEIGHT	x 3.2
CIE SCORE	608.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCERT II

o FOR CONCEPT: #7 Project Management System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Diagnostics			
Documentation	1	2	2
Interactive Support	1	2	2
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	9	3	27
Government Access	5	1	5
Flexibility of Use			
Hardware	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	0	3	0
Output			
DMA Applicable	5	3	15
Understandable	9	2	18
System Resources			
Allocations Required	0	3	0

TOTAL	194.0
NEED WEIGHT	x 3.2
CIE SCORE	620.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIM

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	0	3	0
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	4	3	12
Output			
DMA Applicable	4	3	12
Understandable	6	2	12
System Resources			
Allocations Required	0	3	0

TOTAL	184.0
NEED WEIGHT	x 2.8
CIE SCORE	515.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PRICE

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	5	3	15
Output			
DMA Applicable	4	3	12
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0

TOTAL	190.0
NEED WEIGHT	x 2.8
CIE SCORE	532.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: COCOMO

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	9	2	18
Automated Procedure	1	2	2
Maturity	1	3	3
Vendor Support	1	1	1
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	4	2	8
Training	1	3	3
Output			
DMA Applicable	4	3	12
Understandable	5	2	10
System Resources			
Allocations Required	10	3	30
TOTAL			135.0
NEED WEIGHT			x 2.8
CIE SCORE			378.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIM

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	0	3	0
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	4	3	12
Output			
DMA Applicable	4	3	12
Understandable	6	2	12
System Resources			
Allocations Required	0	3	0

TOTAL	184.0
NEED WEIGHT	x 2.6
CIE SCORE	478.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PRICE

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	5	3	15
Output			
DMA Applicable	4	3	12
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0

TOTAL	190.0
NEED WEIGHT	x 2.6
CIE SCORE	494.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: COCOMO

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	9	2	18
Automated Procedure	1	2	2
Maturity	1	3	3
Vendor Support	1	1	1
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	4	2	8
Training	1	3	3
Output			
DMA Applicable	4	3	12
Understandable	5	2	10
System Resources			
Allocations Required	10	3	30

TOTAL	135.0
NEED WEIGHT	x 2.6
CIE SCORE	351.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SLIM

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	0	3	0
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	4	3	12
Output			
DMA Applicable	4	3	12
Understandable	6	2	12
System Resources			
Allocations Required	0	3	0

TOTAL	184.0
NEED WEIGHT	x 3.2
CIE SCORE	588.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PRICE

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	9	2	18
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	3	1	3
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	5	2	10
Training	5	3	15
Output			
DMA Applicable	4	3	12
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0
TOTAL			190.0
NEED WEIGHT			x 3.2
CIE SCORE			608.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: COCOMO

o FOR CONCEPT: #8 Cost Estimating System

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	9	2	18
Automated Procedure	1	2	2
Maturity	1	3	3
Vendor Support	1	1	1
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	10	1	10
Flexibility of Use			
Hardware	1	2	2
Conceptual Simplicity			
Use	4	2	8
Training	1	3	3
Output			
DMA Applicable	4	3	12
Understandable	5	2	10
System Resources			
Allocations Required	10	3	30

TOTAL	135.0
NEED WEIGHT	x 3.2
CIE SCORE	432.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PERT

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #10 Improved Milestone Identification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	10	2	20
Automated Procedure	1	2	2
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16

TOTAL	163.0
NEED WEIGHT	x 3.0
CIE SCORE	489.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: CPM
- o FOR CONCEPT: #9 Project Path Analysis Method
- o SATISFIES NEED: #10 Improved Milestone Identification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Automated Procedure	3	2	6
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	0	2	0
Training	1	3	3
Output			
DMA Applicable	4	3	12
Understandable	0	2	0
TOTAL			125.0
NEED WEIGHT			x 3.0
CIE SCORE			375.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CPAT

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #10 Improved Milestone Identification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16

TOTAL	173.0
NEED WEIGHT	x 3.0
CIE SCORE	519.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PERT

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	10	2	20
Automated Procedure	1	2	2
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16

TOTAL	163.0
NEED WEIGHT	x 2.8
CIE SCORE	456.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CPM

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Automated Procedure	3	2	6
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	0	2	0
Training	1	3	3
Output			
DMA Applicable	4	3	12
Understandable	0	2	0

TOTAL	125.0
NEED WEIGHT	x 2.8
CIE SCORE	475.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCERT II

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Conceptual Simplicity			
Use	8	2	16
Training	0	3	0
Output			
DMA Applicable	5	3	15
Understandable	9	2	18

TOTAL	158.0
NEED WEIGHT	x 2.8
CIE SCORE	442.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PERT

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	10	2	20
Automated Procedure	1	2	2
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16

TOTAL	163.0
NEED WEIGHT	x 2.6
CIE SCORE	423.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CPM

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Automated Procedure	3	2	6
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	0	2	0
Training	1	3	3
Output			
DMA Applicable	4	3	12
Understandable	0	2	0

TOTAL	125.0
NEED WEIGHT	x 2.6
CIE SCORE	325.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: CPAT
- o FOR CONCEPT: #9 Project Path Analysis Method
- o SATISFIES NEED: #14 Improve Schedule Impact Analysis

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16
TOTAL			173.0
NEED WEIGHT			x 2.6
CIE SCORE			449.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PERT

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	10	2	20
Automated Procedure	1	2	2
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16

TOTAL	163.0
NEED WEIGHT	x 3.2
CIE SCORE	521.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o **IMPLEMENTATION: CPM**

o **FOR CONCEPT: #9 Project Path Analysis Method**

o **SATISFIES NEED: #56 Management Tracking Functions**

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Automated Procedure	3	2	6
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	0	2	0
Training	1	3	3
Output			
DMA Applicable	4	3	12
Understandable	0	2	0

TOTAL	125.0
NEED WEIGHT	x 3.2
CIE SCORE	400.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCERT II

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Conceptual Simplicity			
Use	8	2	16
Training	0	3	0
Output			
DMA Applicable	5	3	15
Understandable	9	2	18

TOTAL	158.0
NEED WEIGHT	x 3.2
CIE SCORE	505.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CPAT

o FOR CONCEPT: #9 Project Path Analysis Method

o SATISFIES NEED: #56 Management Tracking Functions

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	5	2	10
Automated Procedure	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Government Access	10	1	10
Conceptual Simplicity			
Use	5	2	10
Training	1	3	3
Output			
DMA Applicable	8	3	24
Understandable	8	2	16

TOTAL	173.0
NEED WEIGHT	x 3.2
CIE SCORE	553.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: HYPERTEXT

o FOR CONCEPT: #10 Software Eng. Practices Training

o SATISFIES NEED: #18 Faster Intergation of New Empl's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Automated Procedure	10	2	20
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	8	3	24
Conceptual Simplicity Use	10	2	20

TOTAL	127.0
NEED WEIGHT	x 2.2
CIE SCORE	279.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: HYPERTEXT

o FOR CONCEPT: #10 Software Eng. Practices Training

o SATISFIES NEED: #47 Comprehensive Training Program

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Automated Procedure	10	2	20
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	8	3	24
Conceptual Simplicity Use	10	2	20

TOTAL	127.0
NEED WEIGHT	x 3.8
CIE SCORE	482.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: USEIT

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #21 Simulator for Design

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	4	2	8
Diagnostics			
Documentation	5	2	10
Interactive Support	5	2	10
Automated Procedure	8	2	16
Maturity	3	3	9
Vendor Support	7	1	7
Availability	10	3	30
Hardware Compatibility	6	3	18
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	10	2	20
Conceptual Simplicity			
Use	5	2	10
Training	3	3	9
Output			
DMA Applicable	8	3	24
Understandable	8	2	16
System Resources			
Allocations Required	6	3	18

TOTAL	250.0
NEED WEIGHT	x 1.6
CIE SCORE	400.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CS4

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #21 Simulator for Design

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Diagnostics			
Documentation	5	2	10
Interactive Support	1	2	2
Automated Procedure	5	2	10
Maturity	10	3	30
Vendor Support	2	1	2
Availability	10	3	30
Hardware Compatibility	7	3	21
Government Access	7	1	7
Flexibility of Use			
Hardware	8	2	16
Software	8	2	16
Conceptual Simplicity			
Use	7	2	14
Training	8	3	24
Output			
DMA Applicable	7	3	21
Understandable	0	2	0
System Resources			
Allocations Required	0	3	0

TOTAL	222.0
NEED WEIGHT	x 1.6
CIE SCORE	355.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PAWS

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #21 Simulator for Design

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	3	2	6
Diagnostics			
Documentation	8	2	16
Interactive Support	8	2	16
Automated Procedure	6	2	12
Maturity	2	3	6
Vendor Support	3	1	3
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	8	2	16
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	1	3	3
Understandable	1	2	2
System Resources			
Allocations Required	0	3	0

TOTAL	194.0
NEED WEIGHT	x 1.6
CIE SCORE	310.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SHELL

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #22 Program Design Language

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Diagnostics			
Documentation	5	2	10
Interactive Support	3	2	6
Automated Procedure	10	2	20
Maturity	10	3	30
Vendor Support	4	1	4
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	1	2	2
Conceptual Simplicity			
Use	1	2	2
Training	2	3	6
Output			
DMA Applicable	3	3	9
Understandable	8	2	16
System Resources			
Allocations Required	4	3	12

TOTAL	232.0
NEED WEIGHT	x 3.6
CIE SCORE	835.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: USEIT

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #22 Program Design Language

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	4	2	8
Diagnostics			
Documentation	5	2	10
Interactive Support	5	2	10
Automated Procedure	8	2	16
Maturity	3	3	9
Vendor Support	7	1	7
Availability	10	3	30
Hardware Compatibility	6	3	18
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	10	2	20
Conceptual Simplicity			
Use	5	2	10
Training	3	3	9
Output			
DMA Applicable	8	3	24
Understandable	8	2	16
System Resources			
Allocations Required	6	3	18

TOTAL	250.0
NEED WEIGHT	x 3.6
CIE SCORE	900.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CS4

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #22 Program Design Language

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Diagnostics			
Documentation	5	2	10
Interactive Support	1	2	2
Automated Procedure	5	2	10
Maturity	10	3	30
Vendor Support	2	1	2
Availability	10	3	30
Hardware Compatibility	7	3	21
Government Access	7	1	7
Flexibility of Use			
Hardware	8	2	16
Software	8	2	16
Conceptual Simplicity			
Use	7	2	14
Training	8	3	24
Output			
DMA Applicable	7	3	21
Understandable	0	2	0
System Resources			
Allocations Required	0	3	0

TOTAL	222.0
NEED WEIGHT	x 3.6
CIE SCORE	799.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: USEIT

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #57 Software Development Tools

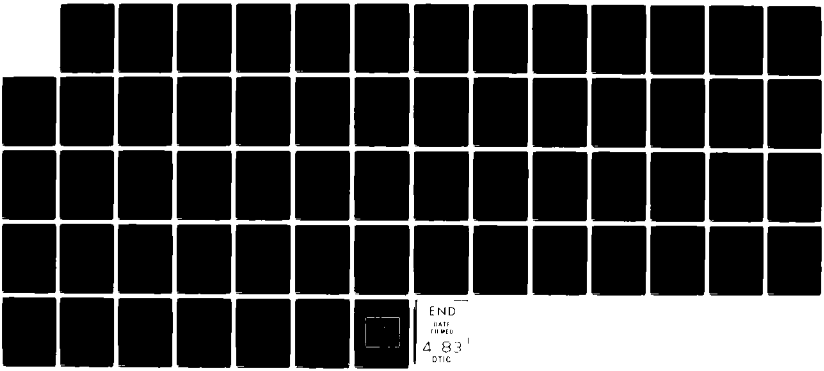
EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	4	2	8
Diagnostics			
Documentation	5	2	10
Interactive Support	5	2	10
Automated Procedure	8	2	16
Maturity	3	3	9
Vendor Support	7	1	7
Availability	10	3	30
Hardware Compatibility	6	3	18
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	10	2	20
Conceptual Simplicity			
Use	5	2	10
Training	3	3	9
Output			
DMA Applicable	8	3	24
Understandable	8	2	16
System Resources			
Allocations Required	6	3	18

TOTAL	250.0
NEED WEIGHT	x 4.2
CIE SCORE	1050.0

AD-A125 626 INTERACTIVE COMPUTER PROGRAM DEVELOPMENT SYSTEM STUDY VOLUME 1(U) GENERAL DYNAMICS FORT WORTH TX FORT WORTH DIV H C CONN ET AL. JAN 83 DMA-2-014-VOL-1

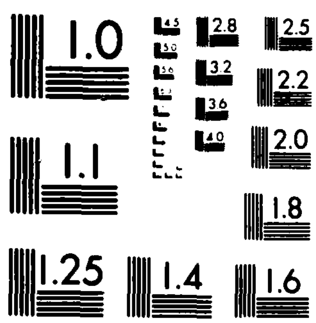
UNCLASSIFIED RADC-TR-83-3-VOL-1 F30602-81-C-0039 F/G 9/2 NL

5/5



END
DATE
FILMED
4 83
DTIC

M-2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PAWS

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	5	3	15
Support Documentation	3	2	6
Diagnostics			
Documentation	8	2	16
Interactive Support	8	2	16
Automated Procedure	6	2	12
Maturity	2	3	6
Vendor Support	3	1	3
Availability	10	3	30
Hardware Compatibility	1	3	3
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	8	2	16
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	1	3	3
Understandable	1	2	2
System Resources			
Allocations Required	0	3	0
TOTAL			194.0
NEED WEIGHT			x 4.2
CIE SCORE			814.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CS4

o FOR CONCEPT: #11 Rapid Prototyping

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Diagnostics			
Documentation	5	2	10
Interactive Support	1	2	2
Automated Procedure	5	2	10
Maturity	10	3	30
Vendor Support	2	1	2
Availability	10	3	30
Hardware Compatibility	7	3	21
Government Access	7	1	7
Flexibility of Use			
Hardware	8	2	16
Software	8	2	16
Conceptual Simplicity			
Use	7	2	14
Training	8	3	24
Output			
DMA Applicable	7	3	21
Understandable	0	2	0
System Resources			
Allocations Required	0	3	0

TOTAL	222.0
NEED WEIGHT	x 4.2
CIE SCORE	932.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

- IMPLEMENTATION: ASET
- FOR CONCEPT: #12 Automated Training Program
- SATISFIES NEED: #12 Improve Manloading

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Conceptual Simplicity Use	8	2	16
System Resources Allocations Required	0	3	0
TOTAL			131.0
NEED WEIGHT			x 2.8
CIE SCORE			366.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: ASET

o FOR CONCEPT: #12 Automated Training Program

o SATISFIES NEED: #18 Faster Intergation of New Empl's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Conceptual Simplicity Use	8	2	16
System Resources Allocations Required	0	3	0

TOTAL 131.0
NEED WEIGHT x 2.8
CIE SCORE 366.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: ASET

o FOR CONCEPT: #12 Automated Training Program

o SATISFIES NEED: #47 Comprehensive Training Program

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	5	3	15
Conceptual Simplicity Use	8	2	16
System Resources Allocations Required	0	3	0

TOTAL	131.0
NEED WEIGHT	x 3.8
CIE SCORE	497.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

- IMPLEMENTATION: SOFTOOL 80
- FOR CONCEPT: #12 Automated Training Program
- SATISFIES NEED: #47 Comprehensive Training Program

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	6	3	18
Support Documentation	5	2	10
Maturity	10	3	30
Availability	10	3	30
Environment Compatibility	4	3	12
Conceptual Simplicity Use	6	2	12
System Resources Allocations Required	0	3	0

TOTAL	112.0
NEED WEIGHT	x 3.8
CIE SCORE	425.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: FAME
- o FOR CONCEPT: #13 Automated Requirements Generation
- o SATISFIES NEED: #1 Formal Requirements Specification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Diagnostics			
Documentation	4	2	8
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	8	3	24
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	9	3	27

TOTAL	307.0
NEED WEIGHT	x 2.6
CIE SCORE	798.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PSL/PSA

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #1 Formal Requirements Specification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	10	2	20
Diagnostics			
Documentation	8	2	16
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	9	3	27
Vendor Support	7	1	7
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	6	2	12
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	5	3	15
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	7	3	21

TOTAL	311.0
NEED WEIGHT	x 2.6
CIE SCORE	808.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SRIMP

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #1 Formal Requirements Specification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	8	2	16
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	10	1	10
Availability	1	3	3
Hardware Compatibility	1	3	3
Environment Compatibility	5	3	15
Government Access	1	1	1
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	4	3	12
TOTAL			229.0
NEED WEIGHT			x 2.6
CIE SCORE			595.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: LARE

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #1 Formal Requirements Specification

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	5	2	10
Diagnostics			
Documentation	4	2	8
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	8	3	24
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	8	3	24
TOTAL			297.0
NEED WEIGHT			x 2.6
CIE SCORE			772.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: RDP 1100

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #5 Requirements Tracking

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	4	2	8
Maturity	1	3	3
Vendor Support	10	1	10
Availability	9	3	27
Hardware Compatibility	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	0	2	0
Training	0	3	0
Output			
DMA Applicable	3	3	9
Understandable	0	2	0
System Resources			
Allocations Required	5	3	15
TOTAL			172.0
NEED WEIGHT			x 2.2
CIE SCORE			378.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FAME

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Diagnostics			
Documentation	4	2	8
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	8	3	24
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	9	3	27

TOTAL	307.0
NEED WEIGHT	x 3.4
CIE SCORE	1043.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: RDP 1100

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	4	2	8
Maturity	1	3	3
Vendor Support	10	1	10
Availability	9	3	27
Hardware Compatibility	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	0	2	0
Training	0	3	0
Output			
DMA Applicable	3	3	9
Understandable	0	2	0
System Resources			
Allocations Required	5	3	15

TOTAL	172.0
NEED WEIGHT	x 3.4
CIE SCORE	584.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SRIMP

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	8	2	16
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	10	1	10
Availability	1	3	3
Hardware Compatibility	1	3	3
Environment Compatibility	5	3	15
Government Access	1	1	1
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	4	3	12

TOTAL	229.0
NEED WEIGHT	x 3.4
CIE SCORE	778.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PSL/PSA

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	10	2	20
Diagnostics			
Documentation	8	2	16
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	9	3	27
Vendor Support	7	1	7
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	6	2	12
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	5	3	15
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	7	3	21

TOTAL	311.0
NEED WEIGHT	x 3.4
CIE SCORE	1057.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: LARE

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	5	2	10
Diagnostics			
Documentation	4	2	8
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	8	3	24
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	8	3	24
TOTAL			297.0
NEED WEIGHT			x 3.4
CIE SCORE			1009.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FAME

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	5	2	10
Diagnostics			
Documentation	4	2	8
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	8	3	24
Vendor Support	8	1	8
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	9	3	27

TOTAL	307.0
NEED WEIGHT	x 4.2
CIE SCORE	1289.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: RDP 1100

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	4	2	8
Maturity	1	3	3
Vendor Support	10	1	10
Availability	9	3	27
Hardware Compatibility	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	0	2	0
Training	0	3	0
Output			
DMA Applicable	3	3	9
Understandable	0	2	0
System Resources			
Allocations Required	5	3	15

TOTAL	172.0
NEED WEIGHT	x 4.2
CIE SCORE	722.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PSL/PSA

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	10	2	20
Diagnostics			
Documentation	8	2	16
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	9	3	27
Vendor Support	7	1	7
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	6	2	12
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	5	3	15
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	7	3	21

TOTAL	311.0
NEED WEIGHT	x 4.2
CIE SCORE	1306.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SRIMP

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	8	2	16
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	7	2	14
Maturity	10	3	30
Vendor Support	10	1	10
Availability	1	3	3
Hardware Compatibility	1	3	3
Environment Compatibility	5	3	15
Government Access	1	1	1
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	4	3	12

TOTAL	229.0
NEED WEIGHT	x 4.2
CIE SCORE	961.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: LARE

o FOR CONCEPT: #13 Automated Requirements Generation

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	8	3	24
Support Documentation	5	2	10
Diagnostics			
Documentation	4	2	8
Interactive Support	6	2	12
Automated Procedure	7	2	14
Maturity	8	3	24
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	8	2	16
Software	10	2	20
Conceptual Simplicity			
Use	8	2	16
Training	6	3	18
Output			
DMA Applicable	7	3	21
Understandable	8	2	16
System Resources			
Allocations Required	8	3	24

TOTAL	297.0
NEED WEIGHT	x 4.2
CIE SCORE	1247.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SDDL

o FOR CONCEPT: #14 Software Design Language

o SATISFIES NEED: #22 Program Design Language

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Diagnostics			
Documentation	4	2	8
Interactive Support	2	2	4
Automated Procedure	4	2	8
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Government Access	10	1	10
Flexibility of Use			
Hardware	7	2	14
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	7	3	21
Understandable	10	2	20
System Resources			
Allocations Required	0	3	0

TOTAL	311.0
NEED WEIGHT	x 3.6
CIE SCORE	1119.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PDL

o FOR CONCEPT: #14 Software Design Language

o SATISFIES NEED: #22 Program Design Language

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	2	3	6
Support Documentation	10	2	20
Diagnostics			
Documentation	0	2	0
Interactive Support	1	2	2
Automated Procedure	2	2	4
Maturity	10	3	30
Vendor Support	0	1	0
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Government Access	5	1	5
Flexibility of Use			
Hardware	6	2	12
Software	8	2	16
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	7	3	21
Understandable	10	2	20
System Resources			
Allocations Required	0	3	0

TOTAL	256.0
NEED WEIGHT	x 3.6
CIE SCORE	921.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SDDL

o FOR CONCEPT: #14 Software Design Language

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Diagnostics			
Documentation	4	2	8
Interactive Support	2	2	4
Automated Procedure	4	2	8
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Government Access	10	1	10
Flexibility of Use			
Hardware	7	2	14
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	7	3	21
Understandable	10	2	20
System Resources			
Allocations Required	0	3	0

TOTAL	311.0
NEED WEIGHT	x 3.4
CIE SCORE	1057.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PDL

o FOR CONCEPT: #14 Software Design Language

o SATISFIES NEED: #41 Organ. Tools/Techniques Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	2	3	6
Support Documentation	10	2	20
Diagnostics			
Documentation	0	2	0
Interactive Support	1	2	2
Automated Procedure	2	2	4
Maturity	10	3	30
Vendor Support	0	1	0
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Government Access	5	1	5
Flexibility of Use			
Hardware	6	2	12
Software	8	2	16
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	7	3	21
Understandable	10	2	20
System Resources			
Allocations Required	0	3	0

TOTAL	256.0
NEED WEIGHT	x 3.4
CIE SCORE	870.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: USER INTERFACE FOR ON-LINE ASSISTANCE

o FOR CONCEPT: #14 Software Design Language

o SATISFIES NEED: #54 Natural Lang. User/Sys. Interface

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	10	2	20
Automated Procedure	10	2	20
Maturity	1	3	3
Vendor Support	0	1	0
Availability	1	3	3
Hardware Compatibility	10	3	30
Environment Compatibility	1	3	3
Government Access	0	1	0
Flexibility of Use			
Hardware	1	2	2
Software	5	2	10
Conceptual Simplicity			
Use	10	2	20
Training	10	3	30
Output			
DMA Applicable	0	3	0
Understandable	10	2	20
System Resources			
Allocations Required	0	3	0

TOTAL	191.0
NEED WEIGHT	x 1.4
CIE SCORE	267.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SDDL

o FOR CONCEPT: #14 Software Design Language

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Diagnostics			
Documentation	4	2	8
Interactive Support	2	2	4
Automated Procedure	4	2	8
Maturity	9	3	27
Vendor Support	9	1	9
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Government Access	10	1	10
Flexibility of Use			
Hardware	7	2	14
Software	10	2	20
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	7	3	21
Understandable	10	2	20
System Resources			
Allocations Required	0	3	0

TOTAL	311.0
NEED WEIGHT	x 4.2
CIE SCORE	1306.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: PDL
- o FOR CONCEPT: #14 Software Design Language
- o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	2	3	6
Support Documentation	10	2	20
Diagnostics			
Documentation	0	2	0
Interactive Support	1	2	2
Automated Procedure	2	2	4
Maturity	10	3	30
Vendor Support	0	1	0
Availability	10	3	30
Hardware Compatibility	7	3	21
Environment Compatibility	8	3	24
Government Access	5	1	5
Flexibility of Use			
Hardware	6	2	12
Software	8	2	16
Conceptual Simplicity			
Use	9	2	18
Training	9	3	27
Output			
DMA Applicable	7	3	21
Understandable	10	2	20
System Resources			
Allocations Required	0	3	0

TOTAL	256.0
NEED WEIGHT	x 4.2
CIE SCORE	1075.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: ADF

o FOR CONCEPT: #14 Software Design Language

o SATISFIES NEED: #5/ Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	0	3	0
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	5	2	10
Maturity	0	3	0
Vendor Support	0	1	0
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	5	3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	10	2	20
Conceptual Simplicity			
Use	4	2	8
Training	4	3	12
Output			
DMA Applicable	5	3	15
Understandable	0	2	0
System Resources			
Allocations Required	9	3	27

TOTAL	163.0
NEED WEIGHT	x 4.2
CIE SCORE	684.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IPF

o FOR CONCEPT: #15 Structured Programming Facility

o SATISFIES NEED: #18 Faster Intergation of New Empl's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	3	2	6
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	6	2	12
Training	0	3	0
System Resources			
Allocations Required	6	3	18

TOTAL	124.0
NEED WEIGHT	x 2.2
CIE SCORE	272.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IPF

o FOR CONCEPT: #15 Structured Programming Facility

o SATISFIES NEED: #34 Automated Text Management System

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	3	2	6
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	6	2	12
Training	0	3	0
System Resources			
Allocations Required	6	3	18

TOTAL	124.0
NEED WEIGHT	x 3.8
CIE SCORE	471.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IPF

o FOR CONCEPT: #15 Structured Programming Facility

o SATISFIES NEED: #55 Modern Source Data Entry Tech's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	3	2	6
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	6	2	12
Training	0	3	0
System Resources			
Allocations Required	6	3	18

TOTAL 124.0
NEED WEIGHT x 4.6
CIE SCORE 570.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: IPF
- o FOR CONCEPT: #15 Structured Programming Facility
- o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	3	2	6
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	6	2	12
Training	0	3	0
System Resources			
Allocations Required	6	3	18

TOTAL	124.0
NEED WEIGHT	x 4.2
CIE SCORE	520.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: UTS 4000 TEXT PROCESSOR

o FOR CONCEPT: #16 Interactive Text Processing

o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	5	2	10
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	6	3	18
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	6	2	12
Training	3	3	9
Output			
DMA Applicable	7	3	21
System Resources			
Allocations Required	9	3	27

TOTAL	185.0
NEED WEIGHT	x 2.0
CIE SCORE	370.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

- **IMPLEMENTATION:** IS/1 INword
- **FOR CONCEPT:** #16 Interactive Text Processing
- **SATISFIES NEED:** #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	8	2	16
Diagnostics			
Documentation	8	2	16
Interactive Support	8	2	16
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	1	2	2
Conceptual Simplicity			
Use	8	2	16
Training	7	3	21
Output			
DMA Applicable	4	3	12
System Resources			
Allocations Required		3	

TOTAL	188.0
NEED WEIGHT	x 2.0
CIE SCORE	376.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o **IMPLEMENTATION: UTS 4000 TEXT PROCESSOR**

o **FOR CONCEPT: #16 Interactive Text Processing**

o **SATISFIES NEED: #34 Automated Text Management System**

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	5	2	10
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	6	3	18
Government Access	5	1	5
Flexibility of Use			
Hardware	5	2	10
Software	5	2	10
Conceptual Simplicity			
Use	6	2	12
Training	3	3	9
Output			
DMA Applicable	7	3	21
System Resources			
Allocations Required	9	3	27

TOTAL	185.0
NEED WEIGHT	x 3.8
CIE SCORE	703.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: IS/1 INword

o FOR CONCEPT: #16 Interactive Text Processing

o SATISFIES NEED: #34 Automated Text Management System

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	8	2	16
Diagnostics			
Documentation	8	2	16
Interactive Support	8	2	16
Maturity	9	3	27
Availability	10	3	30
Hardware Compatibility	3	3	9
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	3	2	6
Software	1	2	2
Conceptual Simplicity			
Use	8	2	16
Training	7	3	21
Output			
DMA Applicable	4	3	12
System Resources			
Allocations Required		3	

TOTAL	188.0
NEED WEIGHT	x 3.8
CIE SCORE	714.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

- o IMPLEMENTATION: The Daily PlanIt
- o FOR CONCEPT: #17 Automated Data Collection
- o SATISFIES NEED: #11 Decreased Paperwork

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Maturity	8	3	24
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	7	3	21
Government Access	5	1	5
Flexibility of Use			
Hardware	4	2	8
Software	10	2	20
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
Output			
DMA Applicable	8	3	24
Understandable	7	2	14
System Resources			
Allocations Required	1	3	3

TOTAL	233.0
NEED WEIGHT	x 2.0
CIE SCORE	466.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: The Daily PlanIt

o FOR CONCEPT: #17 Automated Data Collection

o SATISFIES NEED: #40 Historical Data Base Techniques

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Maturity	8	3	24
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	7	3	21
Government Access	5	1	5
Flexibility of Use			
Hardware	4	2	8
Software	10	2	20
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
Output			
DMA Applicable	8	3	24
Understandable	7	2	14
System Resources			
Allocations Required	1	3	3

TOTAL	233.0
NEED WEIGHT	x 2.6
CIE SCORE	605.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: The Daily PlanIt

o FOR CONCEPT: #17 Automated Data Collection

o SATISFIES NEED: #46 Reduce Accounting Data Report Anomalies

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	8	2	16
Maturity	8	3	24
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	7	3	21
Government Access	5	1	5
Flexibility of Use			
Hardware	4	2	8
Software	10	2	20
Conceptual Simplicity			
Use	7	2	14
Training	7	3	21
Output			
DMA Applicable	8	3	24
Understandable	7	2	14
System Resources			
Allocations Required	1	3	3

TOTAL	233.0
NEED WEIGHT	x 2.8
CIE SCORE	652.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FAVS

o FOR CONCEPT: #19 Software Testing System

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	8	2	16
Diagnostics			
Documentation	7	2	14
Interactive Support	1	2	2
Automated Procedure	10	2	20
Maturity	8	3	24
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	8	3	24
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	8	3	24
Understandable	7	2	14
System Resources			
Allocations Required	8	3	24

TOTAL	307.0
NEED WEIGHT	x 4.2
CIE SCORE	1289.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCMS

o FOR CONCEPT: #19 Software Testing System

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	9	2	18
Diagnostics			
Documentation	6	2	12
Interactive Support	0	2	0
Automated Procedure	9	2	18
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Software	7	2	14
Conceptual Simplicity			
Use	7	2	14
Training	1	3	3
Output			
DMA Applicable	6	3	18
Understandable	9	2	18
System Resources			
Allocations Required	8	3	24

TOTAL	270.0
NEED WEIGHT	x 4.2
CIE SCORE	1134.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FORTRAN 77 ANALYZER

o FOR CONCEPT: #19 Software Testing System

o *SATISFIES NEED: #57 Software Development Tools*

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	10	2	20
Maturity	1	3	3
Availability	5	3	15
Hardware Compatibility	5	3	15
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	6	2	12
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	6	3	18
Understandable	7	2	14
System Resources			
Allocations Required	8	3	24

TOTAL	214.0
NEED WEIGHT	x 4.2
CIE SCORE	898.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SOFTOOL 80

o FOR CONCEPT: #19 Software Testing System

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Diagnostics			
Documentation	8	2	16
Interactive Support	0	2	0
Automated Procedure	8	2	16
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
Output			
DMA Applicable	5	3	15
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0

TOTAL	267.0
NEED WEIGHT	x 4.2
CIE SCORE	1121.4

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FAVS

o FOR CONCEPT: #19 Software Testing System

o SATISFIES NEED: #58 Production Program Optimization

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	8	2	16
Diagnostics			
Documentation	7	2	14
Interactive Support	1	2	2
Automated Procedure	10	2	20
Maturity	8	3	24
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	8	3	24
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	8	3	24
Understandable	7	2	14
System Resources			
Allocations Required	8	3	24

TOTAL	307.0
NEED WEIGHT	x 4.0
CIE SCORE	1228.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FORTRAN 77 ANALYZER

o FOR CONCEPT: #19 Software Testing System

o SATISFIES NEED: #58 Production Program Optimization

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	10	2	20
Maturity	1	3	3
Availability	5	3	15
Hardware Compatibility	5	3	15
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	6	2	12
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	6	3	18
Understandable	7	2	14
System Resources			
Allocations Required	8	3	24

TOTAL	214.0
NEED WEIGHT	x 4.0
CIE SCORE	856.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: SCMS

o FOR CONCEPT: #19 Software Testing System

o SATISFIES NEED: #58 Production Program Optimization

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Support Documentation	9	2	18
Diagnostics			
Documentation	6	2	12
Interactive Support	0	2	0
Automated Procedure	9	2	18
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Software	7	2	14
Conceptual Simplicity			
Use	7	2	14
Training	1	3	3
Output			
DMA Applicable	6	3	18
Understandable	9	2	18
System Resources			
Allocations Required	8	3	24

TOTAL	270.0
NEED WEIGHT	x 4.0
CIE SCORE	1080.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

- IMPLEMENTATION: SOFTOOL 80
- FOR CONCEPT: #19 Software Testing System
- SATISFIES NEED: #58 Production Program Optimization

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	10	3	30
Support Documentation	10	2	20
Diagnostics			
Documentation	8	2	16
Interactive Support	0	2	0
Automated Procedure	8	2	16
Maturity	10	3	30
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	4	3	12
Government Access	5	1	5
Flexibility of Use			
Hardware	7	2	14
Software	7	2	14
Conceptual Simplicity			
Use	6	2	12
Training	8	3	24
Output			
DMA Applicable	5	3	15
Understandable	7	2	14
System Resources			
Allocations Required	0	3	0

TOTAL	267.0
NEED WEIGHT	x 4.0
CIE SCORE	1068.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CAVS

o FOR CONCEPT: #19 Software Testing System

o SATISFIES NEED: #58 Production Program Optimization

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	7	3	21
Support Documentation	8	2	16
Diagnostics			
Documentation	0	2	0
Interactive Support	3	2	6
Automated Procedure	10	2	20
Maturity	1	3	3
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	0	2	0
Software	3	2	6
Conceptual Simplicity			
Use	9	2	18
Training	5	3	15
Output			
DMA Applicable	6	3	18
Understandable	0	2	0
System Resources			
Allocations Required	0	3	0

TOTAL	196.0
NEED WEIGHT	x 4.0
CIE SCORE	784.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FAVS

o FOR CONCEPT: #20 Software Standardization

o SATISFIES NEED: #2 QA Procedures & Guidelines

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	8	2	16
Diagnostics			
Documentation	7	2	14
Interactive Support	1	2	2
Automated Procedure	10	2	20
Maturity	8	3	24
Vendor Support	5	1	5
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	8	3	24
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	8	3	24
Understandable	7	2	14

TOTAL	288.0
NEED WEIGHT	x 3.6
CIE SCORE	1036.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FORTRAN 77 ANALYZER

o FOR CONCEPT: #20 Software Standardization

o SATISFIES NEED: #2 QA Procedures & Guidelines

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	10	2	20
Maturity	1	3	3
Vendor Support	10	1	10
Availability	5	3	15
Hardware Compatibility	5	3	15
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	6	2	12
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	6	3	18
Understandable	7	2	14

TOTAL	200.0
NEED WEIGHT	x 3.6
CIE SCORE	720.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CAVS

o FOR CONCEPT: #20 Software Standardization

o SATISFIES NEED: #2 QA Procedures & Guidelines

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	7	3	21
Support Documentation	8	2	16
Diagnostics			
Documentation	0	2	0
Interactive Support	3	2	6
Automated Procedure	10	2	20
Maturity	1	3	3
Vendor Support	5	1	5
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	0	2	0
Software	3	2	6
Conceptual Simplicity			
Use	9	2	18
Training	5	3	15
Output			
DMA Applicable	6	3	18
Understandable	0	2	0

TOTAL	201.0
NEED WEIGHT	x 3.6
CIE SCORE	723.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FAVS

o FOR CONCEPT: #20 Software Standardization

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	8	2	16
Diagnostics			
Documentation	7	2	14
Interactive Support	1	2	2
Automated Procedure	10	2	20
Maturity	8	3	24
Vendor Support	5	1	5
Availability	10	3	30
Hardware Compatibility	10	3	30
Environment Compatibility	8	3	24
Government Access	10	1	10
Flexibility of Use			
Hardware	5	2	10
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	8	3	24
Understandable	7	2	14

TOTAL	288.0
NEED WEIGHT	x 4.2
CIE SCORE	1209.6

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FORTRAN 77 ANALYZER

o FOR CONCEPT: #20 Software Standardization

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	3	3	9
Support Documentation	0	2	0
Diagnostics			
Documentation	0	2	0
Interactive Support	0	2	0
Automated Procedure	10	2	20
Maturity	1	3	3
Vendor Support	10	1	10
Availability	5	3	15
Hardware Compatibility	5	3	15
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	6	2	12
Software	8	2	16
Conceptual Simplicity			
Use	8	2	16
Training	8	3	24
Output			
DMA Applicable	6	3	18
Understandable	7	2	14

TOTAL	200.0
NEED WEIGHT	x 4.2
CIE SCORE	840.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: CAVS

o FOR CONCEPT: #20 Software Standardization

o SATISFIES NEED: #57 Software Development Tools

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	7	3	21
Support Documentation	8	2	16
Diagnostics			
Documentation	0	2	0
Interactive Support	3	2	6
Automated Procedure	10	2	20
Maturity	1	3	3
Vendor Support	5	1	5
Availability	10	3	30
Hardware Compatibility	5	3	15
Environment Compatibility	6	3	18
Government Access	10	1	10
Flexibility of Use			
Hardware	0	2	0
Software	3	2	6
Conceptual Simplicity			
Use	9	2	18
Training	5	3	15
Output			
DMA Applicable	6	3	18
Understandable	0	2	0

TOTAL	201.0
NEED WEIGHT	x 4.2
CIE SCORE	844.2

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PlanIt Bill-back

o FOR CONCEPT: #21 Chargeback System

o SATISFIES NEED: #48 Chargeback System

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	0	3	0
Support Documentation	0	2	0
Automated Procedure	7	2	14
Maturity	1	3	3
Availability	3	3	9
Hardware Compatibility	10	3	30
Environment Compatibility	5	-3	15
Government Access	5	1	5
Flexibility of Use			
Hardware	1	2	2
Software	10	2	20
Conceptual Simplicity			
Use	0	2	0
Training	0	3	0
Output			
DMA Applicable	6	3	18
Understandable	6	2	12
System Resources			
Allocations Required	3	3	9

TOTAL	137.0
NEED WEIGHT	x 3.4
CIE SCORE	465.8

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: ADA

o FOR CONCEPT: #22 Structured Programming

o SATISFIES NEED: #59 Standardized Phased Development

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Support Documentation	10	2	20
Automated Procedure	1	2	2
Maturity	1	3	3
Availability	2	3	6
Environment Compatibility	3	3	9
Government Access	10	1	10
Flexibility of Use			
Software	1	2	2
Conceptual Simplicity			
Use	6	2	12
Training	2	3	6
Output			
DMA Applicable	9	3	27
Understandable	9	2	18

TOTAL	115.0
NEED WEIGHT	x 3.6
CIE SCORE	414.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FEDSIM

o FOR CONCEPT: #23 User Assistance Function

o SATISFIES NEED: #18 Faster Intergation of New Empl's

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Automated Procedure	1	2	2
Environment Compatibility	10	3	30
Output			
DMA Applicable	10	3	30

TOTAL	65.0
NEED WEIGHT	x 2.2
CIE SCORE	143.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: PEDSIM

o FOR CONCEPT: #23 User Assistance Function

o SATISFIES NEED: #42 User Assistance Function

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Automated Procedure	1	2	2
Environment Compatibility	10	3	30
Output			
DMA Applicable	10	3	30

TOTAL	65.0
NEED WEIGHT	x 3.6
CIE SCORE	234.0

CONCEPT IMPLEMENTATION EVALUATION SHEET

o IMPLEMENTATION: FEDSIM

o FOR CONCEPT: #23 User Assistance Function

o SATISFIES NEED: #44 Error Rate Standards

EVALUATION CRITERIA	EVALUATION	X WEIGHT	= SCORE
Interactive Capability	1	3	3
Automated Procedure	1	2	2
Environment Compatibility	10	3	30
Output			
DMA Applicable	10	3	30

TOTAL	65.0
NEED WEIGHT	x 2.6
CIE SCORE	169.0

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.